

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
INGENIERÍA DE LA SALUD

**Segmentación de tumores cerebrales extra-axiales en imágenes de  
resonancia magnética usando contornos activos**

Segmentation of extra-axial brain tumors in magnetic resonance imaging using  
active contours

Realizado por  
**Gemma Urbanos García**  
Tutorizado por  
**Enrique Nava Baro**  
Departamento  
**Departamento de Ingeniería de Comunicaciones**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, Julio de 2018

Fecha defensa:  
El Secretario del Tribunal



## Resumen

Este Trabajo Fin de Grado consiste en la detección de contornos de tumores cerebrales extra-axiales en imágenes obtenidas por Resonancia Magnética usando técnicas de contornos activos.

Los contornos activos(*Snakes*) son técnicas sofisticadas de detección de formas que se basan en el reajuste de una curva inicial deformable a partir de la minimización de energías externas e internas del contorno.

En primer lugar, se realiza una introducción del problema describiendo la parte biológica sobre tumores cerebrales y la obtención de imágenes por Resonancia Magnética, así como su procesamiento y segmentación.

Posteriormente, se explica la base teórica y matemática de los contornos activos, el conjunto de métodos que son fundamentales para la comprensión y desarrollo de la implementación del algoritmo llevada a cabo en este trabajo.

Finalmente se procede a evaluar el problema, la implementación del método en Matlab y los resultados obtenidos de forma que se puede observar como el algoritmo ofrece los resultados deseados y se demuestra la fiabilidad del estudio a partir de la curva ROC.

**Palabras clave:** Segmentación, conjunto de nivel, contornos activos, tumor, resonancia magnética, ROC.

## **Abstract**

This Final Degree Project consists in detection of extra-axial brain tumors in Magnetic Resonance Imaging using active contours.

Active contours (*Snakes*) are sophisticated forms detection techniques that are based on the readjustment of a deformable initial curve using external and internal energies.

First, there will be an introduction of the problem that describes the biological part about brain tumors and the obtaining of images by magnetic resonance, as well as its processing and segmentation.

Subsequently, it is explained the theoretical and mathematical basis of the active contours, the Level set methods that are fundamental to understand and develop the implementation of the algorithm.

Finally, the problema is evaluated as well as the implementation of the method in Matlab and the results obtained and prove the reliability of the study from the ROC curve.

**Keywords:** Segmentation, *Level Set*, active contours, tumor, magnetic resonance, ROC.

# Índice general

Índice de figuras .....	7
Índice de tablas .....	8
1. Introducción .....	9
1.1. MOTIVACIÓN .....	9
1.2. OBJETIVOS .....	9
1.3. TECNOLOGÍAS UTILIZADAS.....	10
1.4. INTRODUCCIÓN BIOLÓGICA.....	11
1.4.1. Tipos de tumores cerebrales .....	11
1.5. IMÁGENES POR RESONANCIA  MAGNÉTICA(RM).....	16
1.6. FORMATO DICOM .....	18
1.7. SEGMENTACIÓN DE IMÁGENES .....	19
1.7.1  Métodos de segmentación.....	21
1.8. SNAKES O MÉTODO DE CONTORNOS ACTIVOS .....	25
1.8.1  Contornos activos paramétricos .....	26
1.8.2  Contornos activos geométricos.....	31
2. Teoría de contornos activos basados en <i>Level Sets</i> .....	33
2.1. FUNDAMENTOS DE LEVEL SET.....	36
2.2. MÉTODOS BÁSICOS .....	37
2.2.1. El modelo de Mumford y Shah.....	37
2.2.2. Métodos Numéricos: Aproximaciones al problema de Mumford y Shah	39
2.2.3. Modelo Chan-Vese .....	44
2.2.4. Relación entre los modelos Mumford-Shah y Chan-Vese .....	46
2.2.5. Distancia de regularización de la evolución del <i>Level Set</i> (DRLSE)	50

3. Desarrollo y análisis de resultados .....	59
3.1. DESCRIPCIÓN DEL PROBLEMA.....	59
3.2. IMPLEMENTACIÓN DEL MÉTODO .....	59
3.2.1. Estudio del parámetro alpha.....	69
3.2.1.1. Curva ROC.....	73
3.2.2. Cálculo del valor óptimo de alpha.....	82
3.3. RESULTADOS OBTENIDOS.....	91
3.4. CONCLUSIONES.....	92
3.5. FUTURAS LÍNEAS DE INVESTIGACIÓN.....	93
Bibliografía .....	95
Anexos .....	99
ANEXO A. CÓDIGO EN MATLAB .....	99
ANEXO B. IMÁGENES DE REFERENCIA CON ROI.....	115
ANEXO C. REPRESENTACIÓN DE LA CURVA ROC .....	116
ANEXO D. VALORES DE AUC .....	117
ANEXO E. TABLAS OBTENIDAS CON LOS PARÁMETROS NECESARIOS PARA LA CURVA ROC .....	118



# Índice de figuras

Figura 1. Estructura formato DICOM.....	19
Figura 2. Etapas del procesamiento digital de imágenes (Wainschenker & J. Massa, 2011).....	20
Figura 3. Lena con distintos niveles de umbralización .....	22
Figura 4. Segmentación con Split and Merge.....	23
Figura 5. (a) El contorno activo y se deforma en todo momento y en cada punto de con un vector velocidad; (b) el contorno activo y se divide en dos curvas regulares, mientras que $\phi$ sigue como una función.....	34
Figura 6. Valores de $\phi$ en el interior de la región de interés, la zona externa y en el borde del contorno.....	34
Figura 7. Evolución Level Set (Nicoguardo, 2018) .....	35
Figura 8. Casos posibles de localización de la curva C. La energía de minimización ("fitting energy") se minimiza cuando la curva está en el contorno del objeto (Chan & Vese, 2001) .....	46
Figura 9. Diferentes regularizaciones de la función Heaviside(arriba) y la función delta(abajo) (Chan y Vese, 2001) .....	50
Figura 10. a) $\sigma=0.5$ b) $\sigma=1.8$ c) $\sigma=5$ .....	60
Figura 11. a) $hsize=2$ b) $hsize=20$ c) $hsize=50$ .....	61
Figura 12. a) $timestep=-2$ b) $timestep=1$ c) $timestep=5$ .....	61
Figura 13. a) $\lambda=1$ b) $\lambda=10$ c) $\lambda=50$ .....	62
Figura 14. a) $\alpha=-5$ b) $\alpha=5$ c) $\alpha=10$ .....	62
Figura 15. a) $\epsilon=-5$ b) $\epsilon=1$ c) $\epsilon=5$ .....	63
Figura 16. (a) $lter\_inner=2$ (b) $lter\_inner=40$ (c) $lter\_inner=100$ .....	64
Figura 17. (a) $lter\_outer=0$ (b) $lter\_outer=10$ (c) $lter\_outer=50$ .....	64
Figura 18. Dibujo del Level Set cero alrededor del tumor. a) Level Set cero inicial doble b) Level Set cero inicial único. ....	65
Figura 19. Representación de la función Level Set inicial .....	66
Figura 20. Contorno final adaptado al objeto específico .....	66
Figura 21. Representación de la función Level Set Final .....	67
Figura 22. Level Set iniciales no ajustados .....	68



Figura 23. Level Set finales no ajustados.....	68
Figura 24. Level set cero.....	70
Figura 25. Progresión de la curva del snake dependiente del parámetro alpha	71
Figura 26. a) Imagen ROI original b) Canales RGB en el contorno c) Canales RGB fuera del contorno.....	72
Figura 27. a) Imagen ROI original b) Contorno obtenido de la ROI original c) Contorno de la ROI original en binario .....	72
Figura 28. FN, TP, FP, TN de la superposición de contornos .....	74
Figura 29. Superposición de la región deseada y el Snake para cada iteración de alpha .....	75
Figura 30. Curva ROC.....	77
Figura 31. Método para calcular el AUC .....	78
Figura 32. Curva ROC segundo caso .....	80
Figura 33. Comparación de contornos segundo caso .....	81
Figura 34. Curva ROC para valores medios .....	83
Figura 35. Snake obtenido con $\alpha=-2$ en cada caso .....	85
Figura 36. Superposición de la snake con $\alpha=-2$ y el contorno deseado ....	86
Figura 37. Superposición de contornos a) imagen original b) $\alpha=-2$ c) $\alpha=-1$ .....	87
Figura 38. Superposición de contornos a) imagen original b) $\alpha=-2$ c) $\alpha=-1$ .....	87
Figura 39. Snake obtenido con $\alpha=-1$ para cada caso .....	89
Figura 40. Superposición de la snake con $\alpha=-1$ y el contorno deseado ....	90
Figura 41. Imágenes de referencia con ROI .....	115
Figura 42. Curvas ROC obtenidas de cada imagen.....	116

## Índice de tablas

Tabla 1. Clasificación de los tumores según la OMS,2016. ....	12
Tabla 2. Cálculo de AUC.....	79
Tabla 3. Valores medios de sensibilidad y especificidad por cada iteración de alpha .....	82
Tabla 4. Valores de AUC obtenidos para cada imagen.....	117





# 1. Introducción

## 1.1. MOTIVACIÓN

La principal motivación de este trabajo es conseguir que a partir del método de contornos activos generar una curva que se adapte correctamente al contorno de la zona que se quiere segmentar, en este caso, el tumor cerebral extra-axial de las imágenes de los casos clínicos dados.

Actualmente, el profesional en medicina tiene que seleccionar la región de interés (*region of interest*, ROI) manualmente, y esto puede dar lugar a inexactitudes ya que la topología del tumor puede ser más o menos heterogénea, mostrando más o menos protuberancias, lo cual complica que el dibujo sea exacto y se adapte bien a los bordes del tumor.

El método de contornos activos puede solucionar este problema, ya que sólo sería necesario que el profesional indicara alrededor del tumor un rectángulo aproximado al mismo, y automáticamente, el programa ajusta esta curva a los contornos del tumor.

Este estudio puede proporcionar al profesional en medicina una técnica que facilite su trabajo, evitando los errores humanos y pudiendo ayudar al diagnóstico.

## 1.2. OBJETIVOS

El principal objetivo de este trabajo es, precisamente, la misma que la motivación al mismo, la detección de contornos en tumores extra-axiales en imágenes de resonancia magnética a partir de contornos activos.

Se trata de utilizar un algoritmo concreto basado en contornos activos que aproxime lo mejor y más eficientemente posible el contorno del tumor cerebral. Para ello, se cuenta con unas imágenes dadas por un profesional sanitario, en



las que se han marcado las ROIs, de forma que se usan estas imágenes como referencia para demostrar si el estudio funciona de forma correcta y se aproxima a la región objetivo.

En este trabajo se realizará un estudio de la metodología utilizada para llevar a cabo este contorno activo, haciendo hincapié en la susceptibilidad de este según como le afectan los parámetros que se definen según la base matemática de estos métodos.

Por último, se validarán las soluciones a partir de una comparación estadística de los resultados que da el programa y la imagen con el ROI que ha sido proporcionada por el profesional médico la cual se ha tomado como referencia.

### **1.3. TECNOLOGÍAS UTILIZADAS**

El trabajo se ha desarrollado a partir del software de MATLAB.

MATLAB es un entorno de programación con un lenguaje propio (M), diseñado para el análisis iterativo y con un lenguaje que expresa las matemáticas y los arrays directamente (MathWorks, 1994-2018).

Se ha elegido este software debido a su capacidad de procesamiento, análisis y desarrollo de algoritmos de imágenes con la herramienta Image Processing Toolbox. MATLAB ofrece una gran variedad de funciones para el tratamiento de imágenes, destacando entre ellas la capacidad de leer ficheros DICOM (*Digital Imaging and Communications in Medicine*) ya que las imágenes médicas que se han proporcionado en este trabajo tienen esta extensión. MATLAB permite tratar este tipo de imágenes con funciones precisas de forma clara y sencilla.

Además, el método de contornos activos que se utiliza tiene una base matemática robusta, y este programa es una buena herramienta para expresar el lenguaje matemático intuitivamente, lo cual ayuda al desarrollo del algoritmo y su interpretación.



## 1.4. INTRODUCCIÓN BIOLÓGICA

Los seres vivos están formados por células. En los adultos, estas células se renuevan cuando se necesitan reemplazar otras viejas o dañadas. Los jóvenes, además de renovar células dañadas, crean nuevas células para completar su desarrollo.

Un tumor se define como una masa de tejido de una parte del organismo cuyas células sufren un crecimiento anormal y no tienen ninguna función fisiológica; estas células tienen tendencia a invadir otras partes del cuerpo (Real Academia Española[RAE], 2017).

Un tumor cerebral es una masa de células innecesarias que crecen en el cerebro o en el conducto vertebral. Existen dos tipos básicos de tumores cerebrales: los primarios y los metastásicos. Los tumores cerebrales primarios se originan en el propio cerebro y tienden a permanecer en el mismo. Los tumores cerebrales metastásicos se originan en otra parte del organismo y en algún momento acaban extendiéndose hacia el cerebro (Pérez Segura, 2017).

Para describir el grado de malignidad o agresividad de un tumor cerebral los usan las palabras “benigno” o “maligno”. Las diferencias entre ambos radican en que un tumor cerebral benigno tiene los bordes definidos, sus células crecen muy lentamente y no suele propagarse. Los tumores cerebrales malignos, en cambio, crecen rápidamente, son invasivos y potencialmente mortales. No es sencillo clasificar un tumor como “benigno” o “maligno” ya que pueden afectar múltiples factores además de lo que se ha observado al diagnosticar el tumor (ABTA, 2012).

### 1.4.1. Tipos de tumores cerebrales

La OMS (Organización Mundial de la Salud) ha creado un sistema de clasificación de los tumores primarios del sistema nervioso central basado en la célula de origen del proceso tumoral y el grado histológico que define su agresividad (Sinning, 2017).



*Tabla 1. Clasificación de los tumores según la OMS,2016.*

	DIAGNÓSTICO OMS 2016	GRADO OMS
GLIOMAS DE BAJO GRADO		
Astrocitoma difuso	Astrocitoma difuso IDH mutado	Grado II
	Astrocitoma difuso IDH nativo	Grado II
	Astrocitoma difuso NOS	Grado II
Oligodendroglioma	Oligodendroglioma IDH mutado y 1p/19q codeletado	Grado II
	Oligodendroglioma NOS	Grado II
Oligoastrocitoma	Oligoastrocitoma NOS	Grado II
GLIOMAS ALTO GRADO		
Astrocitoma anaplásico	Astrocitoma anaplásico IDH mutado	Grado III
	Astrocitoma anaplásico IDH nativo	Grado III
	Astrocitoma anaplásico NOS	Grado III
Oligodendroglioma anaplásico	Oligodendroglioma anaplásico IDH mutado y 1p/19q codeletado	Grado III



	Oligodendroglioma anaplásico NOS	Grado III
Oligoastrocitoma anaplásico	Oligoastrocitoma NOS	Grado III
Glioblastoma	Glioblastoma IDH mutado	Grado IV
	Glioblastoma IDH nativo	Grado IV
	Glioblastoma NOS	Grado IV
MEDULOBLASTOMA		
Meduloblastoma genéticamente definido	Meduloblastoma SHH activado y p53 mutado	Grado IV
	Meduloblastoma SHH activado y p53 nativo	Grado IV
	Meduloblastoma no WNT no SHH	Grado IV
Meduloblastoma histológicamente definido	Meduloblastoma clásico	Grado IV
	Meduloblastoma desmoplástico/nodular	Grado IV
	Meduloblastoma con nodularidad extensa	Grado IV



	Meduloblastoma de células grandes/anaplásico	Grado IV
	Meduloblastoma NOS	Grado IV

Según su grado histológico los tumores pueden clasificarse como:

- Grado I – Tumores delimitados, crecen despacio, no se extienden, sus células se asemejan a las normales y generalmente se pueden extraer por cirugía.
- Grado II - Tumores de borde difuso, crecen despacio, pero pueden extenderse. Con tendencia a transformarse en tumores de un grado más alto.
- Grado III - Tumores infiltrantes de rápido crecimiento que pueden extenderse. Las células no se asemejan a las células normales.
- Grado IV – Tumores de rápido crecimiento y metastásicos. Son difíciles de tratar.

El glioma es un tipo de tumor que se produce en el cerebro y la médula espinal. Los gliomas se llaman así ya que son tumores que comienzan en las células gliales, que son células viscosas de sostén que rodean las neuronas e influyen en su funcionamiento.

Hay tres tipos de células gliales que pueden producir tumores. Los gliomas se clasifican también dependiendo del tipo específico de células que más se parecen a sus características histológicas. Hay tres tipos principales de gliomas (Pérez Segura, 2017):

- Ependimomas: Se forman a partir de las células ependimarias, que son las encargadas de producir líquido cefalorraquídeo y revisten los ventrículos, el encéfalo y el centro de la médula espinal.
- Astrocitomas: Los astrocitomas se llaman así porque nacen de células llamadas astrocitos. Aparecen en el parénquima cerebral y no suelen ser metastásicos.



- Oligodendrogliomas: Surgen de los oligodendrocitos, células que cubren las fibras nerviosas del cerebro.

Otros ejemplos de tumores cerebrales importantes son:

- Glioblastoma (GB): El glioblastoma o glioblastoma multiforme es un astrocitoma de grado IV, es decir, es el más maligno. Suele tener presencia de necrosis y derrames sanguíneos alrededor del tumor.
- Meduloblastoma: El meduloblastoma es un tumor de grado IV que suele aparecer en el cerebelo y que tiende a extenderse a través del líquido cefalorraquídeo y la médula espinal.
- Meningioma: Los meningiomas son tumores de las meninges y suelen ser de grado I, es decir, benignos, por lo que no se extienden muy rápidamente. Sin embargo, los meningiomas pueden crecer si no se tratan a tiempo y convertirse en tumores de grado II o III. Son los más frecuentes, más del 30% de los tumores cerebrales son meningiomas.
- Oligoastrocitomas y Oligodendrogliomas: Son tumores muy poco frecuentes de grado II o III que se diferencian en dos tipos de tumores distintos, Oligoastrocitomas y oligodendrogliomas.

Para poder diagnosticar y evaluar un tumor cerebral se usa las pruebas de imagen por tomografía axial computarizada(TC) o la resonancia magnética nuclear, que además de ayudar al médico a identificar la ubicación exacta del tumor, puede dar información sobre su estado.

Las imágenes por resonancia magnética (*magnetic resonance imaging*, MRI) se consideran la mejor forma para detectar tumores en estas áreas ya que las imágenes son más precisas que las de una tomografía computarizada. Sin embargo, no muestran los huesos del cráneo tan bien como las tomografías, por lo que es posible que no se muestren los efectos de los tumores en el cráneo.



## **1.5. IMÁGENES POR RESONANCIA MAGNÉTICA(RM)**

La imagen por resonancia magnética es una técnica no invasiva que produce imágenes tridimensionales de la parte del cuerpo a estudiar, usándose para el diagnóstico y seguimiento del paciente. A diferencia de la tomografía axial computarizada(TC), no usa radiación ionizante, sino campos magnéticos, por lo que el paciente no se expone a radiaciones dañinas.

Para obtener imágenes por RM se introduce al paciente en el seno de gran campo magnético de forma que los protones se alinean con el campo. Una vez magnetizado, se envía una serie de pulsos en forma de ondas de radiofrecuencia a través del paciente, los protones se estimulan y luchan contra este campo magnético girando fuera del equilibrio de manera que esta energía es absorbida y posteriormente devuelta y recogida por unos sensores de imagen de resonancia magnética y se reconstruye la imagen, en planos sagitales, axiales, coronales y cualquier tipo de plano oblicuo.

Siendo más específicos, la imagen por resonancia magnética se basa en la ionización de átomos que tienen un número impar de electrones en su núcleo, concretamente se habla de resonancia magnética basada en hidrógeno. La imagen por RM se obtiene después de haber sometido a los núcleos de hidrógeno a un campo magnético.

Cuando el campo magnético incide sobre estos átomos estos se orientan paralela o antiparalelamente y rotando sobre sí mismos en forma de spin. Algo más de la mitad de los protones se alinean paralelos al campo, y será esta proporción extra de núcleos alineados paralelamente la responsable de generar momentos magnéticos.

Además del movimiento del spin, los protones sometidos al campo magnético realizan un giro cónico, como una peonza. Al estar sometido a la gravedad, da lugar a un movimiento de precesión con una geometría cónica de su movimiento. La frecuencia de este movimiento de precesión depende del núcleo que se esté





estudiando y es directamente proporcional al campo magnético al que se está sometiendo el núcleo. Esto se llama Ley de Larmor:

$$w = \gamma B_0 \quad (1)$$

Siendo  $w$  la frecuencia de precesión,  $B_0$  el campo magnético y  $\gamma$  una constante llamada razón giromagnética.

Si se le aplica una onda de radiofrecuencia idéntica a la frecuencia de precesión, los núcleos entran en resonancia y absorben la energía de este pulso de radiofrecuencia. Una vez que cesa la absorción de energía los núcleos vuelven a su estado anterior y devuelven la energía siguiendo una curva de tiempo determinada por la relación de estos núcleos con otros campos magnéticos del exterior.

Cabe destacar que, si se somete a un paciente en un gran campo magnético y se emiten ondas de radiofrecuencia, todo el organismo entrará en resonancia, lo que se traduce en un mal resultado de la prueba. Es por ello por lo que se utiliza un gradiente dentro del campo, para que sólo un plano tenga la frecuencia de precesión adecuada.

Las imágenes en RM se utilizan ampliamente en el diagnóstico y seguimiento de tumores cerebrales debido a su alta sensibilidad y detalle. Sin embargo, las técnicas clásicas de RM no son del todo exactas y ofrecen poca información sobre el estado fisiológico del tumor. Por ello, actualmente se están usando técnicas como la espectroscopia o la perfusión y difusión, para determinar el grado histológico del tumor, monitorización a la hora de tratarlo y otras funcionalidades que están ahora mismo en estudio.

La información que se obtiene por RM es procesada y transformada en imágenes. Estas imágenes suelen tener un formato estándar DICOM y pueden ser tratadas y analizadas de distintas formas. En el caso de este trabajo, las imágenes son tratadas con técnicas de segmentación y detección de formas. A continuación, se introducirá en que consiste el formato DICOM y algunas



técnicas de segmentación que se pueden realizar en la imagen, destacando el método de contornos activos, que es en el que se basa este trabajo.

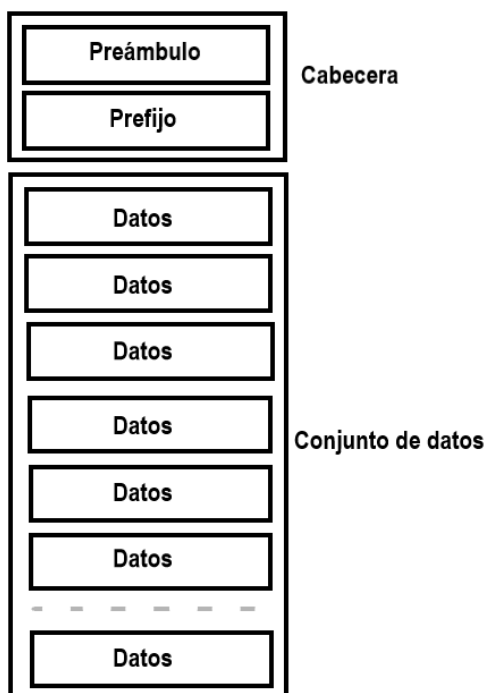
## 1.6. FORMATO DICOM

DICOM (*Digital Imaging and Communication in Medicine*) es un estándar de comunicación entre sistemas de información y a la vez un formato de almacenamiento de imágenes médicas que ha posibilitado el intercambio de información entre dispositivos debido a que se ha implantado como un formato estándar en las imágenes médicas y es mundialmente conocido y aceptado. El formato DICOM ha sido impulsada por ACR y NEMA, que siguen siendo organizaciones privadas y por ello no todos los hospitales tienen porque tratar con imágenes con formato DICOM (Mildenger, Eichelberg, & Martin, 2002).

En la Figura 1 se puede ver la estructura del formato DICOM, el cual es como un contenedor. Posee dos partes, la cabecera y los datos de la imagen (píxeles).

La cabecera posee la información sobre el paciente, el sistema de adquisición o modalidad, los parámetros de la imagen, es decir, el conjunto de datos referente a la medicina. La cabecera tiene un identificador único con sintaxis de transferencia UID y está formada por un preámbulo de 128 bytes que se usa para la compatibilidad con otros ficheros, seguido de un prefijo de 4 bytes.

Por otro lado, está formado por el conjunto de datos, que contiene la imagen o imágenes especificadas, las cuales vienen codificadas y comprimidas en diversos estándares como JPEG, JPEG 2000, JPEG y además tiene la siguiente información: una etiqueta de identificación, un valor de representación que indica el tipo del dato almacenado, la longitud del dato y el propio dato cuyo contenido depende de la etiqueta, pueden ser los datos del paciente, la prueba realizada... (Forero, Sarmiento, & Díaz, 2001)



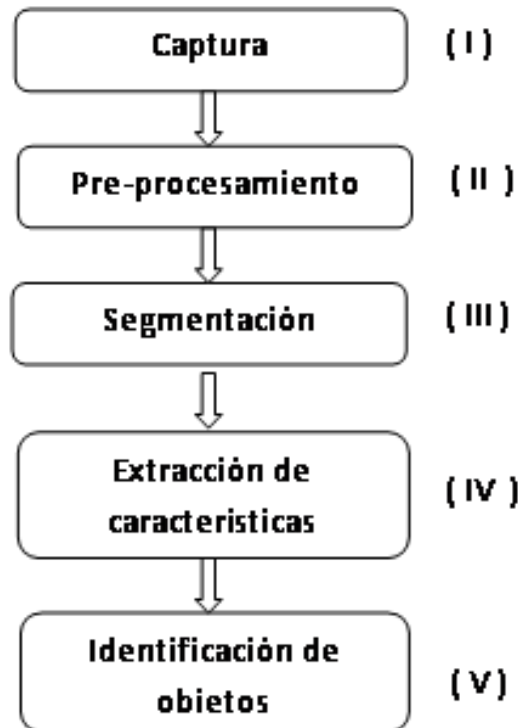
*Figura 1. Estructura formato DICOM*

## 1.7. SEGMENTACIÓN DE IMÁGENES

Una imagen puede ser definida como una función en el espacio bidimensional (2D) de intensidad de luz  $f(x, y)$ , donde  $x$  e  $y$  representan las coordenadas espaciales y la función  $f$  es el nivel de gris correspondiente a ese punto. Una imagen digital puede considerarse como una matriz cuyos índices de fila y de columna identifican un punto en imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Cada uno de estos elementos se llama píxel (Palomino & Concha, 2009).

Para poder transformar una imagen a imagen digital, obtener información de esta imagen digital o mejorarla, se necesita su procesamiento. El procesamiento de imágenes digitales permite muchas funciones sobre las imágenes, algunos ejemplos de ello son los filtros espaciales, transformaciones de intensidad, restauración de imágenes, segmentación, detección de formas... Estos dos últimos ejemplos son en los se basa este trabajo.

El procesamiento de imágenes tiene varias etapas, las cuales se muestran en la Figura 2,



*Figura 2. Etapas del procesamiento digital de imágenes (Wainschenker & J.Massa, 2011)*

La primera etapa es la captura, donde se adquieren las imágenes, para lo cual se necesitan un sensor y un digitalizador de forma que el objeto pueda convertirse en una señal digital.

La etapa II es el preprocesamiento de la imagen, para detectar fallos y eliminarlos de distintas formas, ya sea eliminando ruidos, suavizando contornos, detectar bordes...en esta etapa se incluye el proceso y filtrado para mejorar la imagen.

La tercera etapa es la segmentación, en la que se divide la imagen en regiones o segmentos de interés con unas propiedades concretas.

El paso siguiente es la extracción de características, también conocido como representación y descripción, se basa en el estudio e identificación del objeto segmentado.

La última etapa, en la que se identifican los objetos se basa en reconocimiento e interpretación de estos con procesos automáticos (Wainschenker & J. Massa, 2011).



## 1.7.1 Métodos de segmentación

La segmentación de imágenes que se basa en dividir la imagen en varias partes u objetos, siendo estos grupos de píxeles el objeto de interés y el resto el fondo. La división de estas regiones se caracteriza ya que cada región presenta unas características similares tales como el nivel de gris, la intensidad o la textura. Cada imagen tiene unas características propias y serán mejor segmentadas por métodos determinados. En el caso de las imágenes médicas, por ejemplo, una imagen del cerebro tendrá unos requerimientos distintos a una del corazón. Además, una imagen médica puede tener ruido, volúmenes parciales, artefactos... luego cada una tendrá un requerimiento de segmentación.

### 1.7.1.1. Umbralización

La umbralización (*thresholding*) se basa en la segmentación de imágenes a partir de un valor de intensidad llamado umbral(*threshold*), que separa objetos deseados. Es la técnica más básica de segmentación. Lo más utilizado es la binarización por umbral, es decir, siendo la imagen original  $f(x,y)$ , la imagen umbralizada  $g(x, y)$ , se fija un umbral de la siguiente forma (Palomino & Concha, 2009),

$$\begin{cases} g(x, y) = 255, & \text{si } f(x, y) > \text{Umbral} \\ g(x, y) = 0, & \text{si } f(x, y) \leq \text{Umbral} \end{cases}$$

De esta forma se tiene una imagen en blanco y negro, sin tonalidades de grises de por medio, que permite diferenciar bien los objetos del fondo.

Para elegir el umbral se utiliza el histograma de la imagen, de forma que se puede escoger el nivel de gris más conveniente para el objeto que se desea segmentar. Es una técnica efectiva cuando el objeto a estudiar tiene unas tonalidades homogéneas y diferenciables, pero no es eficaz en imágenes tridimensionales y no puede aplicarse a imágenes multicanal.



*Figura 3. Lena con distintos niveles de umbralización*

#### **1.7.1.2. Segmentación orientada a regiones**

La segmentación de una imagen orientada a regiones se basa en un proceso que divide la imagen en  $n$  subregiones, de forma que estas regiones deben cumplir una serie de propiedades, que definen el test de homogeneidad de una imagen:

- Cada píxel debe ser único por región.
- La unión de todas las subregiones debe formar la imagen completa.
- Los píxeles de cada región deben estar conectados.
- Los píxeles de las regiones que no son la región objetivo deben cumplir que las características de los píxeles de la región a segmentar y, además, dos regiones adyacentes han de ser distintas.

En este tipo de segmentación destacan dos métodos principales: crecimiento de regiones y división y fusión de regiones (Gonzalez, 2016).

#### **1.7.1.3. Crecimiento de regiones**

La técnica de crecimiento de regiones consiste en agrupar píxeles que reúnen alguna característica similar y formar regiones. Partiendo de un conjunto de píxeles en las distintas regiones de la imagen, llamados semillas, las regiones se van expandiendo incluyendo aquellos píxeles que se detecta que pertenecen a esa región. Para cada píxel detectado, se comprueba si cumple la regla de homogeneidad, es decir, si todos los píxeles de la nueva región siguen teniendo propiedades similares, en cuyo caso se amplía la región añadiéndole ese píxel. El procedimiento acaba cuando no queda ningún píxel que añadir.



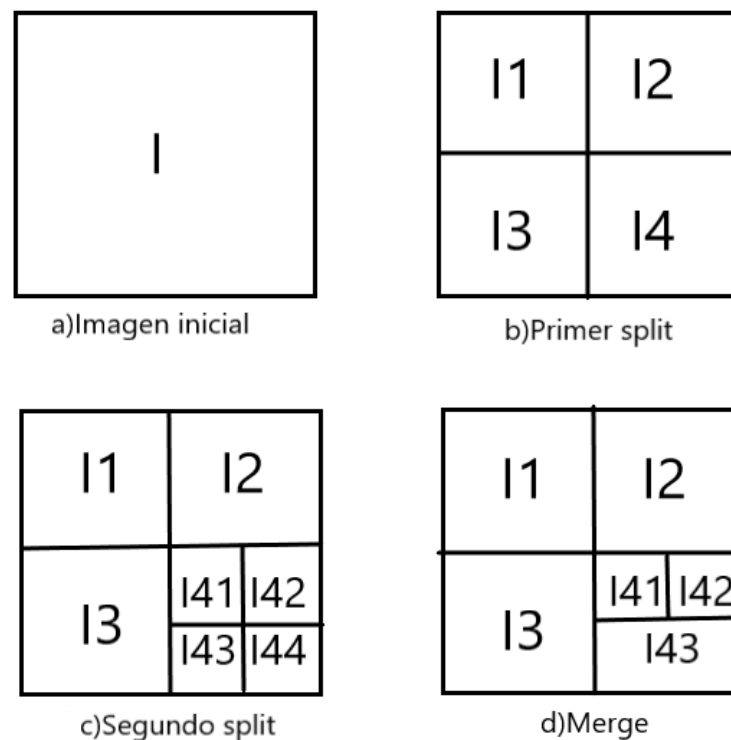
Las semillas pueden ser escogidas manualmente o de forma automática, y la elección de estas semillas hace necesaria la fusión de algunas regiones, las cuales deben tener propiedades similares.

#### 1.7.1.4. División y fusión de regiones (Split and merge)

Este método consiste en la división de la imagen inicialmente en un conjunto de regiones que se escogen arbitrariamente, y aplicando el test de homogeneidad las regiones adyacentes se fusionarán o no.

Partiendo de una imagen inicial  $I$ , se divide la imagen en cuatro regiones, cada subimagen que no sea homogénea se vuelve a dividir en otras cuatro subimágenes y así sucesivamente hasta que todas las regiones son homogéneas. Una vez realizadas las particiones de las imágenes se realizan las fusiones entre las regiones que cumplan el test de homogeneidad.

En la Figura 4 se observa en un ejemplo de cómo funciona el método.



*Figura 4. Segmentación con Split and Merge*



#### 1.7.1.5. Watershed

Una imagen en escala de grises puede ser vista como un relieve topográfico, donde cada relieve se corresponde con el nivel de gris de un píxel. Los píxeles con mayor intensidad forman los límites de las regiones.

El agua se encuentra entre las cuencas de los píxeles con distintos niveles de gris, y cuando el agua alcanza el borde de la cuenca, las regiones adyacentes se fusionan.

#### 1.7.1.6. Agrupamiento(*clustering*)

El “*clustering algorithm*” o algoritmo de agrupación, trata de dividir un conjunto de datos de entrada en subconjuntos (*clusters*), de manera que los elementos de cada subconjunto compartan características similares, y su agrupamiento sea llevado a cabo según similitud o cercanía. Entre los métodos de *clustering* destaca el método *k-means*, que consiste en fijar un número de *clusters* y asignar un *cluster* a cada elemento para después recalcular los *clusters* a partir de los datos asignados hasta que se consigue la convergencia, por ejemplo, que los píxeles no cambien de *cluster* (Coto, 2003).

#### 1.7.1.7. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (*Artificial Neural Network- ANN*) son redes paralelas de procesamiento de elementos o nodos que simulan el aprendizaje del cerebro humano. Cada neurona en la red se corresponde a un píxel en una imagen de entrada. Estas neuronas sufren estímulos externos (el color del píxel) y estímulos locales, que proceden de la conexión de una neurona con sus neuronas vecinas. Los estímulos se acumulan hasta que superan el umbral y se producen impulsos en la salida. Este método es muy robusto frente al ruido y a las variaciones del parámetro de entrada. El uso más frecuente en segmentación de imágenes es como clasificador, con un conjunto de datos de entrada suficientes para que la red neuronal pueda aprender (Olabe, 1998).





#### **1.7.1.8. Modelos deformables**

Los contornos activos o modelos deformables son técnicas basadas en la diferencia de energías externas e internas de forma que, fijada una curva inicial, esta se va deformando para acabar ajustándose al contorno deseado. Estos métodos se basan en encontrar el contorno del objeto que se quiere segmentar con la mayor minimización de energía posible. Su principal ventaja es que genera curvas cerradas directamente, siendo muy robusto frente al ruido, pero se necesita insertar manualmente la curva inicial aproximada para que el modelo se adapte al contorno (Mitiche & Ben Ayed, 2011). Este método es en el que se va a basar este trabajo, y a continuación se explica más detalladamente.

### **1.8. SNAKES O MÉTODO DE CONTORNOS ACTIVOS**

Los contornos activos o Snakes se definen como una curva deformable influenciada por energías externas e internas. Al minimizarse la energía la curva se mueve y se adapta al contorno que se quiere extraer.

Hay que tener en cuenta que un snake no encuentra de forma automática el contorno de una imagen, si no que mejora una curva que se ha creado inicialmente cercana al contorno del objeto (Kass, Witkin, & Terzopoulos, 1988).

El modelo de Snakes ha sido modificado y mejorado por múltiples autores realizando distintas formas de minimización de la energía con métodos como diferencias finitas, elementos finitos, algoritmos voraces, programación dinámica, redes neuronales... (Vese & Guyader, 2015)

Los contornos activos pueden ser contornos activos paramétricos y geométricos. La diferencia entre uno y otro es que los paramétricos se representan únicamente como curvas parametrizadas y esta curva evoluciona según los cambios de intensidad en el nivel de gris en los bordes de la imagen. Por otro lado, los contornos activos geométricos están basados en la evolución de la curva a partir de Level Sets, y dibujan el contorno buscando la homogeneidad de distintas características dentro de una región de la imagen.



### 1.8.1 Contornos activos paramétricos

Los contornos activos paramétricos clásicos fueron creados por Kass, Witkin y Terzopoulos y se basan en la minimización de un funcional de energía, que es mínimo cuando la curva está en los bordes del objeto, de forma que el snake se ajusta al borde utilizando esta información.

La curva del snake evoluciona a partir de fuerzas internas y externas hasta acercarse lo máximo posible a un mínimo local. A partir de esto, se pueden definir las energías que actúan sobre el snake (Kass, Witkin, & Terzopoulos, 1988).

- Energía interna: En este tipo de energía se distinguen,
- Energía de contorno: Procura que la curva sea continua, actúa como una fuerza que otorga resistencia a que la curva sea encogida y/o estirada, es decir, marca la elasticidad de la curva.
- Energía de curvatura: Es la que suaviza la curva. Es una fuerza que provoca que la curva se resista a plegarse o torcerse, es decir, marca la rigidez de la curva.
- Energía externa: Es la que se encarga de aproximar la curva a los mínimos locales del objeto deseado. Se suele usar el gradiente de la imagen como función.

Las fuerzas internas suavizan la curva y las fuerzas externas son las que hacen que la curva se aproxime al mínimo local del objetivo de segmentación.

Por lo tanto, la energía total que actúa sobre la imagen puede definirse de la siguiente forma,

$$E_{TOTAL} = \alpha E_{contorno} + \beta E_{curvatura} + \gamma E_{externa} \quad (2)$$

Siendo  $\alpha$ ,  $\beta$  y  $\gamma$  parámetros que priorizan una energía sobre la otra, ya que dependiendo de las imágenes unas energías deben tener más peso que otras.

Un valor muy alto de  $\alpha$  hará que la curva evolucione rápidamente al contorno deseado, pero puede ocasionar que se encoja demasiado y la curva pueda llegar a colapsar.



El término  $\beta$  marca la suavidad de la curva, por lo tanto, un valor demasiado alto de este parámetro provocará un snake demasiado suave y no será capaz de detectar esquinas.

El término  $\gamma$  es el que se encarga de aproximar la curva a los mínimos locales, si se elige un valor demasiado alto, la curva puede no ajustarse a los límites del objeto deseado.

Particularizando algo más en los términos de energía interna y externa, se puede decir que,

$$E_{interna} = E_{contorno} + E_{curvatura} \quad (3)$$

Es decir, esta energía controla sus características intrínsecas. Sin embargo, la energía externa, es la que guía al snake hacia los bordes del objeto de interés. La energía externa también está compuesta por dos tipos de energía,

$$E_{externa} = E_{imagen} + E_{externa\ restrictiva} \quad (4)$$

Siendo la energía de la imagen la que atrae al snake al contorno deseado y la energía externa restrictiva se introduce manualmente y debe fijar el snake cerca del contorno objetivo, de forma que la curva pueda evolucionar a la forma deseada (Vélez, Sánchez, Moreno, & Esteban, 2005).

La energía de la imagen tiene en cuenta tres funcionales: de línea, borde y terminación de forma que puede definirse como en la ecuación 5.

$$E_{imagen} = w_{línea}E_{línea} + w_{borde}E_{borde} + w_{terminación}E_{terminación} \quad (5)$$

Donde  $w$  es el peso que se le da a cada una de estas energías y que determinará la evolución de la curva. A su vez, estos funcionales de energía se definen como:



- Funcional de línea: Está influenciado por la intensidad de los píxeles de la imagen. El signo de  $w_{línea}$  determina si el *snake* es atraído por las líneas con mayor o menor intensidad. Por ejemplo, en una imagen con 256 tonalidades de gris, el valor de  $w_{línea}=1$  hace que el *snake* converja hacia las líneas más oscuras y  $w_{línea}=-1$  hacia las más claras. La definición de este funcional es simple y se define en la ecuación 6,

$$E_{línea} = I(x, y) \quad (6)$$

Donde  $I(x, y)$  es una imágenes con píxeles colocados en las coordenadas  $x, y$  en escala de grises.

- Funcional de borde: El *snake* es dirigido hacia al borde de interés, se define como en la ecuación 7,

$$E_{borde} = -|\nabla I(x, y)|^2 \quad (7)$$

Donde  $\nabla I(x, y)$  es el gradiente de una imagen. La operación gradiente es capaz de detectar grandes cambios de intensidad, por lo que los bordes más finos puede perderlos. Para poder solucionar esto se usa el operador Laplaciano, que detecta más detalles y líneas delgadas, a costo de que el funcional será más sensible al ruido. El Laplaciano se define como en la ecuación 8,

$$|\nabla I(x, y)|^2 = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (8)$$

La forma de atraer al *snake* hacia el contorno de interés es a partir de la función Gaussiana con una desviación estándar  $\sigma$ . Incluyendo la Gaussiana a la ecuación 9, se hace una convolución con la imagen.



$$E_{borde} = -|\nabla[G_{\sigma}(x, y) * I(x, y)]|^2 \quad (9)$$

Una gran desviación estándar produce una borrosidad que puede provocar que aumente la zona a detectar, pero los contornos no se detecten bien, por lo tanto, lo que se suele hacer es que primero el contorno funcione con una borrosidad alta hasta que se halle cerca del contorno objetivo y disminuir la desviación, de forma que el *snake* se adapte de forma precisa.

- Funcional de terminación: Esta energía permite detectar terminaciones de líneas y esquinas por lo que se necesita una función Gaussiana con desviación para que abarque una mayor zona alrededor del contorno objetivo.

$$C(x, y) = G_{\sigma}(x, y) * I(x, y) \quad (10)$$

En una esquina o terminación de una línea, hay un cambio brusco en el ángulo del gradiente. Este ángulo se calcula de la siguiente forma:

$$\theta = \arctan \left[ \frac{\frac{\partial C}{\partial y}}{\frac{\partial C}{\partial x}} \right] \quad (11)$$

Pudiendo calcular del mismo el vector tangente y normal a la dirección del gradiente

$$n = (\cos \theta, \sin \theta); \quad n_{\perp} = (-\sin \theta, \cos \theta) \quad (12)$$

Siendo  $n$  el vector tangente y  $n_{\perp}$  el vector normal. A partir de estos vectores se puede definir la energía de terminación como,



$$E_{terminación} = \left[ \frac{\partial n_{\perp}}{\partial \theta} \right]^{-1} = \frac{\frac{\partial^2 C}{\partial n_{\perp}^2}}{\frac{\partial C}{\partial n}} \quad (13)$$

La energía de terminación mide cuanto cambia el ángulo del gradiente, al ser un cambio muy brusco en esquinas, se utiliza la inversa del vector normal para que se minimice la energía.

Por otro lado, la energía externa restrictiva de la ecuación número 4, se define, como se ha dicho anteriormente como la curva inicial que introduce el usuario. Se proponen dos tipos de energías externas restrictivas:

- Energía externa de atracción o de muelle. A partir de dos puntos, uno localizado en el snake y otro en la imagen, guía al snake hacia el contorno del objeto deseado. La ecuación de esta energía se corresponde con la energía del muelle en física mecánica,

$$E_{externa \text{ de atracción}} = k|x_1 - x_2|^2 \quad (14)$$

donde  $x_1$  es un punto en la imagen,  $x_2$  es un punto ubicado en el snake y  $k$  es la constante de rigidez (similar a la constante de Hooke).

- Energía externa de repulsión o energía de volcán: esta energía hace que el snake se aleje de una región de la imagen, lo cual sirve para poder alejar el snake de ciertos mínimos locales que no interesan para que pueda orientarse hacia el contorno objetivo.

$$E_{externa \text{ de repulsión}} = \frac{k}{|x_1 - x_2|^2} \quad (15)$$



### 1.8.2 Contornos activos geométricos

Los modelos geométricos de contornos activos están basados en la evolución de curvas por flujos geométricos. La curva se deforma con una velocidad que depende de unos parámetros geométricos intrínsecos (regularidad de la curva) y de la adaptación al contorno de la imagen (Osher & Sethian, 1988).

Este modelo se basa en los métodos de conjunto de nivel o Level Set, de forma que el Level Set cero de una función de Level Set (LSF) es el contorno que evolucionará basándose en unas propiedades matemáticas que se explican a lo largo del siguiente capítulo. Los contornos basados en Level Set son siempre cerrados y suaves, proporcionando flexibilidad a la segmentación.

Los métodos basados en contornos activos geométricos más destacados son los métodos variacionales, que intentan minimizar el funcional de energía. Un ejemplo es el método Chan y Vese en 1999, que propusieron un método variacional añadiendo una restricción al funcional de energía de Level Set, haciendo que su modelo pueda detectar contornos automáticamente (Chan & Vese, 1999).

A continuación, se explican los fundamentos de Level Sets, ya que este método es la base de este trabajo.







## 2. Teoría de contornos activos basados en *Level Sets*

El método de conjunto de nivel (*Level Set Method*, LSM) fue creado como una técnica matemática que permite el reconocimiento de formas utilizando una función con signo. Se utiliza normalmente para la segmentación de imágenes en 2D o en 3D (Osher & Sethian, 1988).

Siendo  $\gamma$  una curva cerrada en el dominio de imagen  $\Omega$ , dicha curva se deforma y converge para delimitar las regiones de segmentación deseadas. El conjunto de nivel cero (*Zero Level Set*,  $\Gamma$ ) es la curva actual, es decir, un contorno cerrado que se va definiendo dependiendo de la función del *Level Set* (LSF). La ecuación del movimiento de  $\Gamma$  sometida a una velocidad  $\mathbf{v}$ , sirve para definir la evolución de la curva. La velocidad a la que está sometida tiene la misma dirección de su normal en cada punto tal y como se representa en la Figura 5. Esta velocidad puede depender de la posición, el tiempo, la geometría y las propiedades externas de la imagen (Mitiche & Ben Ayed, 2011).

$$\mathbf{V} = V\mathbf{n} \quad (16)$$

Las velocidades pueden ser de tres tipos.

Tipo 1.  $V$  es una función de curvatura del *Level Set* cero.

Tipo 2.  $V$  es de la forma  $\{F, n\}$  donde  $F$  es un campo vectorial que depende de la posición y posiblemente del tiempo, pero no de la curva.

Tipo 3.  $V$  es una función escalar que depende de la posición y el tiempo.

El LSM utiliza la curva de nivel cero en la función del *Level Set* ( $\Phi$ ), de forma que esta función toma valores positivos dentro del contorno  $\Gamma$ , valores negativos fuera y cero en los límites, tal y como se representa en la Figura 6. Si se aplica una ecuación de movimiento en el *Level Set* cero se obtiene un flujo que determinará



la evolución de la curva y esta se ajustará el contorno. Sus ventajas es que no depende de ningún parámetro y permite el cambio de topología.

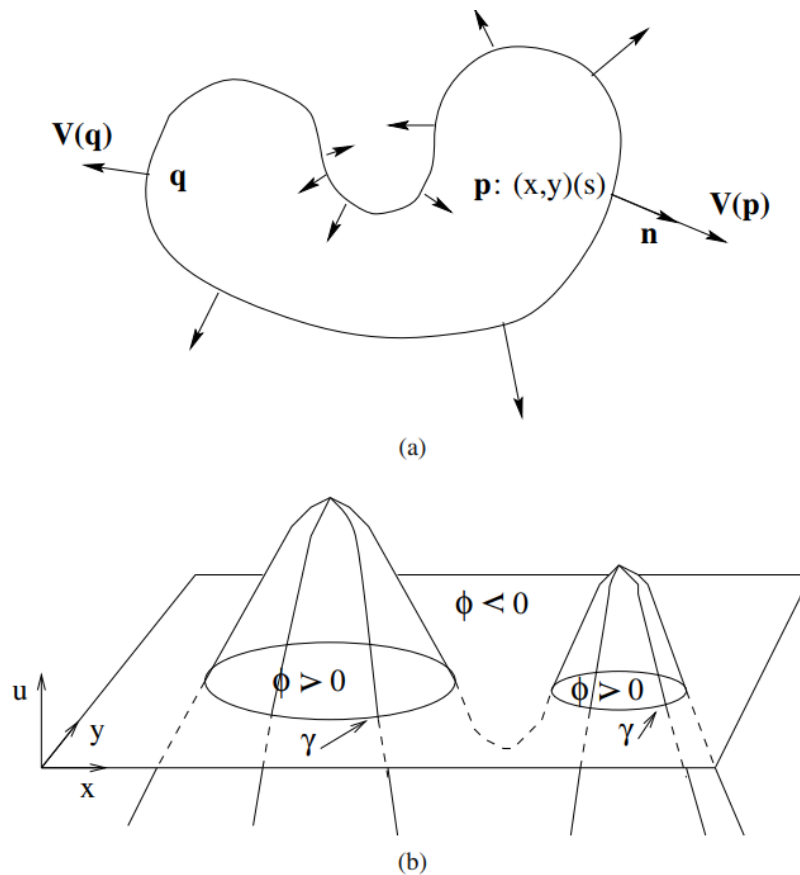


Figura 5. (a) El contorno activo y se deforma en todo momento y en cada punto de con un vector velocidad; (b) el contorno activo y se divide en dos curvas regulares, mientras que  $\phi$  sigue como una función (Mitiche & Ben Ayed, 2011)

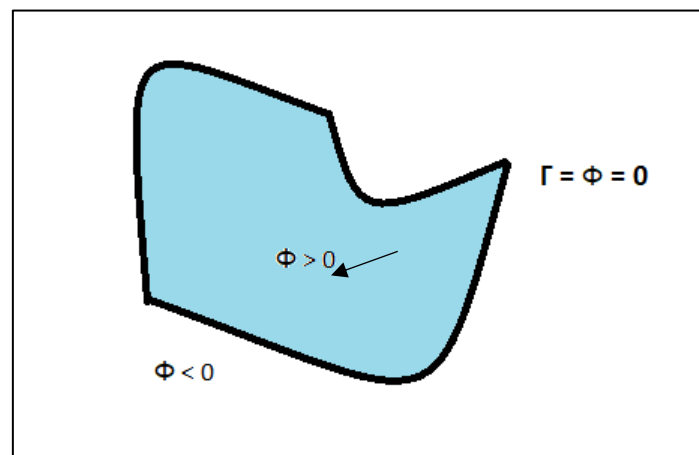
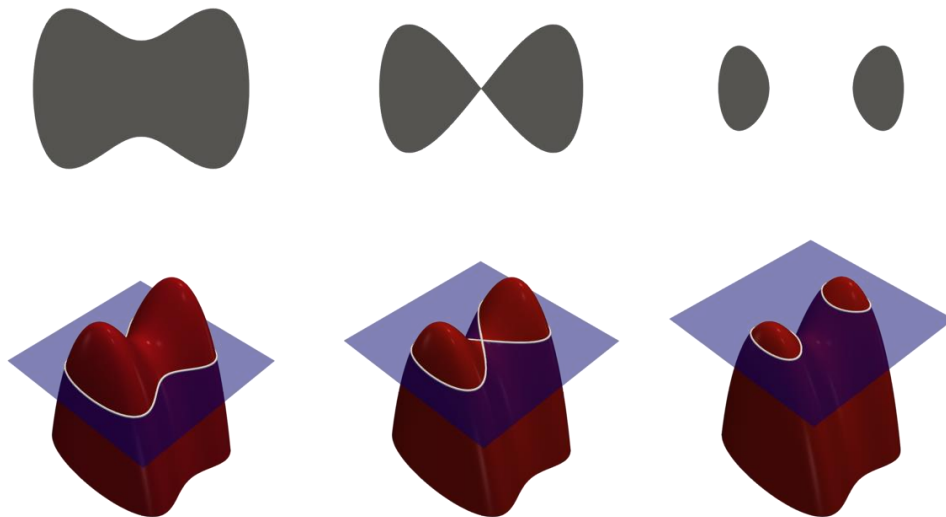


Figura 6. Valores de  $\phi$  en el interior de la región de interés, la zona externa y en el borde del contorno



En la Figura 7 se ve claro el comportamiento del modelo. Siendo el volumen rojo  $\phi$ , la superficie azul el plano xy, la figura gris el conjunto de píxeles donde  $\phi > 0$  y el borde de esta figura el *Level Set* cero ( $\phi=0$ ). Si se quiere detectar cuando el contorno que se ha dividido en dos, dejando fijo el plano xy, se va desplazando el objeto hacia abajo hasta que se encuentra cuando se divide.



*Figura 7. Evolución Level Set (Nicoguardo, 2018)*

Si se realiza un muestreo de la curva y su movimiento se define con un algoritmo de descenso, las curvas cerradas que delimitan la región segmentada pueden dividirse o fusionarse durante su evolución. Estos cambios en la topología de curva ofrecen una gran ventaja en casos de segmentación difícil.

Los principales tipos de métodos de conjunto de nivel se clasifican en métodos de banda estrecha (*narrow band*) o de campo disperso (*sparse field*) (Lamas Rodríguez, Quesada Barriuso, Argüello, Heras, & Bóo, 2012).

- La idea del método de banda estrecha es crear una banda delgada que se vaya adaptando a los elementos vecinos, y realizar el cálculo solo en estos puntos. Se introduce complejidad de programación, pero se ahorra computacionalmente.

- El método de campo disperso actualiza los elementos alrededor del contorno en cada iteración del algoritmo.



## 2.1. FUNDAMENTOS DE LEVEL SET

Sea  $\Gamma$  el conjunto de curvas planas suaves cerradas, simples y regulares, y siendo  $\gamma$  una función, donde  $\gamma: s \in [0,1] \rightarrow \gamma(s) \in \Omega$ . Un contorno activo  $\gamma$  se representa como el *Level Set* cero de una función  $\varphi: \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ :

$$\forall s, t \quad \varphi(x(s, t), y(s, t), t) = 0 \quad (2.41) \quad (17)$$

La derivada de la ecuación 17 con respecto al tiempo es, suponiendo que  $\varphi$  es suficientemente suave,

$$\frac{d\varphi}{dt} = \frac{\partial \varphi}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial \varphi}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial \varphi}{\partial t} = \left( \nabla \varphi, \frac{\partial \gamma}{\partial t} \right) + \frac{\partial \varphi}{\partial t} = 0 \quad (18)$$

Debido a  $\frac{\partial \varphi}{\partial t} = V\mathbf{n}$  se tiene,

$$\frac{\partial \varphi}{\partial t} = -V(\nabla \varphi, \mathbf{n}) \quad (19)$$

A la vez también se cuenta con,

$$\forall s \in [0,1] \quad \frac{d\varphi}{ds} = \frac{\partial \varphi}{\partial x} \frac{\partial x}{\partial s} + \frac{\partial \varphi}{\partial y} \frac{\partial y}{\partial s} = \left( \nabla \varphi, \frac{\partial \gamma}{\partial s} \right) = 0 \quad (20)$$

Siendo  $\frac{\partial \gamma}{\partial s}$  la tangente a la curva en  $s$  y  $\nabla \varphi$  la normal de la curva. Considerando que  $\mathbf{n}$  está orientada hacia fuera y  $\varphi$  es positiva dentro del *Level Set* cero, se deduce la ecuación 21,

$$\mathbf{n} = \frac{\nabla \varphi}{\|\nabla \varphi\|} \quad (21)$$



Sustituyendo la ecuación 21 en 19 se obtiene en la ecuación 25 la evolución temporal de  $\varphi$ ,

$$\frac{\partial \varphi}{\partial t} = V \|\nabla \varphi\| \quad (22)$$

Para las velocidades del primer tipo, es decir,  $V$  es una función de la curvatura de la curva evolutiva, la curvatura se da en términos de la función de *Level Set* mediante:

$$k = \operatorname{div} \left( \frac{\mathbf{n}}{\|\mathbf{n}\|} \right) = -\operatorname{div} \left( \frac{\nabla \varphi}{\|\nabla \varphi\|} \right) = \frac{\varphi_{xx}\varphi_y^2 - 2\varphi_x\varphi_y\varphi_{xy} + \varphi_{yy}\varphi_x^2}{(\varphi_x^2 + \varphi_y^2)^{\frac{3}{2}}} \quad (23)$$

La evolución del contorno sólo se produce en los puntos donde LSF es igual a cero. Para poder llevar a cabo esta evolución se definen velocidades de extensión, implementadas a través de bandas estrechas de forma que la velocidad en un punto sea la misma que la velocidad del punto más cercano a él perteneciente al contorno activo. Para ello se necesita que el contorno inicial se cruce con el contorno objetivo.

Una ventaja del LSM es que es fácil acceder a la región deseada de la segmentación ya que el signo de  $\varphi$  determina qué puntos están dentro del contorno y cuáles están fuera (Mitiche & Ben Ayed, 2011).

## 2.2. MÉTODOS BÁSICOS

### 2.2.1 El modelo de Mumford y Shah

David Mumford y Jayant Shah, propusieron un problema que permite calcular aproximaciones una imagen inicial  $I$ . Estas aproximaciones se basan en dividir  $I$  en trozos. Su modelo trata de minimizar la energía para mejorar las aproximaciones, suavizando de imágenes y detección de límites para su



segmentación (Mumford & Shah, Boundary detection by minimizing functionals, 1985).

Su modelo ha sido estudiado en profundidad, sobre todo en el análisis de imágenes y la visión por computador, tanto es así que la mayoría de las otras formulaciones se pueden explicar cómo variaciones de esta.

### 2.2.1.1. Definición

Siendo  $\Omega \subset \mathbb{R}^d$  el dominio de la imagen,  $I: \Omega \rightarrow \mathbb{R}$  es una imagen en una o dos o tres dimensiones en escala de grises. Para poder segmentar la imagen se debe descomponer  $\Omega$  del dominio de la imagen  $I$  tal que

$$\Omega = \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_n$$

Generalmente, se supone que  $\Omega$  está abierto, limitado y conectado. Con ello se puede observar que la imagen  $I$  varía suavemente y / o lentamente dentro de cada  $\Omega_i$  o varía de forma discontinua y / o rápida a lo largo de la mayor parte del límite  $K$  entre diferentes  $\Omega_i$  (Bar, y otros, 2011).

La formulación de Mumford-Shah busca una aproximación suave  $M: x = (x, y) \in \Omega \subset \mathbb{R}^2 \rightarrow M(x) \in \mathbb{R}$  de la imagen original  $I: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  y un conjunto  $K$  de discontinuidades, para representar los límites de la imagen, minimizando el funcional (Mitiche & Ben Ayed, 2011).

$$F_{MS}(M, K) = \int_{\Omega} (I - M)^2 dx + \rho \int_{\frac{\Omega}{K}} \|\nabla M\|^2 dx + \lambda \rho l(K) \quad (24)$$

Donde  $\rho$  y  $\lambda$  son constantes positivas,  $l$  es la longitud, y  $\nabla M = (M_x + M_y) = \left(\frac{\partial M}{\partial x}, \frac{\partial M}{\partial y}\right)$  es el gradiente espacial de  $M$ .

El primer término lleva la función a una región cerca de la imagen original. El segundo término se utiliza para aquellas imágenes que son muy heterogéneas, para ello se le aplica una suavidad a través de los bordes. El último término se relaciona con la longitud de los límites regulares.



Simplificando el funcional para aquellas imágenes constantes en todas sus partes, siendo  $R_k$  la partición determinada por  $K$  y  $M_k$  aquellos valores de aproximación de  $I$  dentro de una región. Se trata de fijar una partición y un conjunto de valores  $M_k$ .

$$M(x) = M_k \quad \forall x \in R_k \quad (25)$$

$$F_{MS}(M, K) = \sum_k \int_{R_k} (I - M_k)^2 dx + \lambda l(K) \quad (26)$$

## 2.2.2 Métodos Numéricos: Aproximaciones al problema de Mumford y Shah

El método de Mumford y Shah es un problema bien formulado, pero presenta un problema de regularización en el término libre, para solucionarlo se han realizado muchos estudios que intentan aproximar este método. En este trabajo se explican tres de ellos, la interpretación Bayesiana, la implementación graduada de no convexidad y el funcional de Ambrosio y Tortorelli.

### 2.2.2.1. Interpretación Bayesiana

La segmentación de una imagen puede establecerse como una estimación Bayesiana máxima a posteriori (MAP) que determina, en todas las particiones posibles del dominio de una imagen, una partición de una probabilidad de MAP (Mitiche & Ben Ayed, 2011).

Sea  $f$  una imagen degradada conocida y  $u$  la imagen original, la estimación de MAP de  $u$  es el valor más probable de  $u$  dada  $f$  (ecuación 27).

$$\arg \max_u P(u|f) \quad (27)$$



Donde  $P(u|f)$  es la probabilidad condicional de  $u$  dada  $f$ . Aplicando el teorema de Bayes, que liga la probabilidad de  $f$  dado  $u$  con la probabilidad de  $u$  dado  $f$ , se obtiene la ecuación 28,

$$P(u|f) = \frac{P(f|u)P(u)}{P(f)} \quad (28)$$

Siendo  $P(f|u)$  la función de probabilidad de distribución del ruido (PDF). El término  $P(u)$  proporciona información de la región a segmentar. Este término introduce restricciones geométricas en la solución, cuya longitud se define en la ecuación 29,

$$P(u) \propto e^{-\alpha l(K)} \quad (29)$$

donde  $\alpha$  es una constante positiva.

Para simplificar la ecuación 28, se aplica  $-\log$ , resultando la ecuación 30,

$$\begin{aligned} \max_u P(u|f) &\leftrightarrow \max_u \{P(u)P(u|f)\} \\ &\leftrightarrow \min_u \{-\log P(u) - \log P(f|u)\} \end{aligned} \quad (30)$$

En esta expresión, el primer término  $-\log P(u)$  actúa como una regularización. El segundo término  $-\log P(f|u)$  se relaciona con el deterioro que produce  $f$  en  $u$  (Vese & Guyader, 2015).

Teniendo esto en cuenta, la estimación Bayesiana se convierte en el siguiente problema de minimización (ecuación 31 y 32), del que la mayoría de los algoritmos de segmentación variacional optimizan una variante.

$$F_{MS}(M, K) = \sum_k \int_{R_k} (I - M_k)^2 dx + \lambda l(K) \quad (31)$$





$$\arg \min_u \sum_k \int_{x \in R_k} -\log P(f | \mathbf{R}_k) dx - \log P(u) \quad (32)$$

El primer término es un término de datos que mide la probabilidad de la imagen dentro de las regiones de partición. Cada región  $R_k$  se caracteriza por un conjunto de parámetros que se han estimado al hacer la partición. El modelo de región que suele usarse es la distribución gaussiana (ecuación 33),

$$P(f | \mathbf{R}_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(f-\mu_k)^2}{2\sigma_k^2}} \quad (33)$$

Donde el conjunto de parámetros de la región  $R_k$  es  $\{\mu_k, \sigma_k\}$ , siendo, respectivamente,  $\mu_k$  y  $\sigma_k$ , la media y la desviación estándar de los datos de la imagen dentro de  $R_k$ .

Teniendo en cuenta la información que proporciona la ecuación 33 sobre las particiones deseadas, si se toma una misma  $\sigma_k$  en todas las regiones, es decir,  $\sigma_k = \sigma \forall k$ , y si se establece  $\alpha = \frac{\lambda}{\sigma^2}$ , la estimación bayesiana reduce la minimización de la función por partes constantes de Mumford-Shah (Mitiche & Ben Ayed, 2011).

#### 2.2.2.2. Implementación graduada de no convexidad

La implementación graduada de no convexidad es un método de continuación basado en iteraciones para resolver una serie de problemas, primero resuelve un problema fácil utilizando una versión convexa con  $F_{MS}$  utilizando los mínimos globales de la función ya que un mínimo global único sirve como una aproximación inicial. Para simplificar, el método es a menudo descrito para imágenes unidimensionales. Este método intenta minimizar el funcional de Mumford y Shah, cuyo problema con las imágenes unidimensionales es determinar una aproximación suave  $h$  de una función  $h_0$  definida en un intervalo real  $\omega$  y un conjunto de discontinuidades  $K$ , siendo



$$F_{MS}(h, K) = \int_{\omega} (h - h_0)^2 dx + \rho \int_{\omega/K} h'^2 dx + \lambda \rho \text{card}(K) \quad (34)$$

Donde card denota la cardinalidad. Discretizando  $F_{MS}$  se obtiene,

$$F_{MS}^0(h, K) = \sum_i^n (h^i - h_0^i)^2 + f^0(h^i - h^{i-1}) \quad (35)$$

Donde  $h^i$  es el valor en el punto  $i$  de  $h$  en los puntos discretos  $i=1, 2, \dots, n$ . A su vez, en  $\Omega$ ,  $f^0$  es definido por la ecuación 36, y  $K$  es el conjunto de puntos que engloba (ecuación 37).

$$f^0(t) = \begin{cases} t^2 & \text{si } t \leq 1 \\ 1 & \text{si } t > 1 \end{cases} \quad (36)$$

$$K = \{i | |h^i - h^{i-1}| > 1\} \quad (37)$$

Este funcional no es convexo y puede tener varios mínimos locales, luego no sirve para hacer una aproximación inicial. Considerando la siguiente versión de  $F_{MS}$ :

$$F_{MS}^p(h, k) = \sum_i^n (h^i - h_0^i)^2 + f^p(h^i - h^{i-1}) \quad (38)$$

Donde  $f^p$  está definida por,



$$f^p(t) = \begin{cases} t^2, & \text{si } |t| < 1 \\ 1 - \frac{(|t| - r)^2}{4p}, & \text{si } \frac{1}{r} < |t| \leq r \\ 1, & \text{si } r \leq |t| \end{cases} \quad (39)$$

$$r = \sqrt{4p + 1}$$

Se puede demostrar que  $f^p$  es convexo cuando  $p \geq 1$  y, por lo tanto, existe un mínimo global de  $f^1$ . Así, el proceso de graduación no convectivo buscará un mínimo disminuyendo  $p$  de 1 a 0.

En resumen, la implementación no graduada de convexidad es un método de continuación que itera dos pasos, el primero minimiza una versión amplia del funcional y el siguiente usa el resultado como una inicialización para minimizar la anterior versión (Mitiche & Ben Ayed, 2011).

### 2.2.2.3. Funcional Ambrosio y Tortorelli

El funcional Ambrosio-Tortorelli es una aproximación del método Mumford-Shah que ha sido muy utilizado para la segmentación de imágenes. Se basa en una secuencia de problemas variacionales elípticos más simples, de forma que reemplazan el contorno del objeto por una función 2D. Este método tiene una fácil implementación, pero se ha observado que dependiendo de los valores de los parámetros la segmentación varía y a veces no llega a converger al funcional original de Mumford-Shah (Yu & Huang, 2017).

Una de las formas de expresar el funcional de Mumford-Shah es,

$$E[u, \Gamma] = \frac{\alpha}{2} \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + \beta H^1(\Gamma) + \frac{\gamma}{2} \int_{\Omega} (u - g)^2 dx, \quad (40)$$

Donde  $\Omega \subset \mathbb{R}^2$ ,  $\alpha, \beta$  y  $\gamma$  son parámetros positivos,  $g$  es el nivel de gris de la imagen,  $u$  es la imagen que se quiere segmentar,  $\Gamma$  es el contorno del objeto en la imagen, al minimizar,  $\nabla u$  es pequeño en  $\Omega \setminus \Gamma$ , y  $\Gamma$  es lo más pequeño posible ya que  $u$  está cerca de  $g$ . Por último,  $H^1(\Gamma)$  es la medida de Hausdorff en una



dimensión, esto es, la cantidad de puntos en el conjunto si el conjunto es finito o  $\infty$  si el conjunto es infinito. Esta medida es prácticamente imposible de calcular, así que para eludirlo Ambrosio y Tortorelli regularizan el funcional proponiendo la formulación de la ecuación 41,

$$\begin{aligned} AT_{\epsilon}[u, \varphi] = & \frac{\alpha}{2} \int_{\Omega} (\varphi^2 + k_{\epsilon}) |\nabla u|^2 dx \\ & + \beta \int_{\Omega} (\epsilon |\nabla \varphi|^2 + \frac{1}{4\epsilon} (1 - \varphi)^2) dx \\ & + \frac{\gamma}{2} \int_{\Omega} |u - g|^2 dx, \end{aligned} \quad (41)$$

dónde  $\epsilon$  es el parámetro de regularización positivo,  $k_{\epsilon}$  es un parámetro usado para evitar que el funcional se vuelva degenerado, y  $\varphi$  es una nueva variable desconocida que idealmente es una aproximación de la función de salto de  $u$ , por ejemplo:

$$\varphi(x) \equiv \begin{cases} 0, & \text{si } x \in S_u \\ 1, & \text{si } x \notin S_u \end{cases} \quad (42)$$

El funcional regularizado de Ambrosio y Tortorelli es muy utilizado en estudios de ciencia e ingeniería para la simulación de límites y grietas en cristales dendríticos, hidrodinámica de fluidos simples o en fracturas (Yu & Huang, 2017).

### 2.2.3. Modelo Chan-Vese

El modelo creado por Chan y Vese es un caso particular del método Mumford Shah basado en un funcional minimizado que detecta los contornos en las imágenes, a los que se les llama  $u_0$ . (Chan & Vese, 2001).

#### 2.2.3.1. Descripción del método

Sea  $\Omega$  un subconjunto abierto y acotado de  $\mathbb{R}^2$  y  $\partial\Omega$  su frontera, ( $\Omega$  suele estar en el plano). Tomando  $u_0$  una imagen dada (suele tomar valores entre 0 y 255) y definida como una función de  $\Omega$  en  $\mathbb{R}$ . Sea  $C$  la curva de evolución inicial que



puede situarse en cualquier lugar de la imagen y puede tomar cualquier forma excepto nula, esta curva permite una mayor velocidad a la hora de que la curva converja ya que da una forma aproximada inicial. Y, por último,  $c_1$  y  $c_2$  dos constantes que representan la media en el interior y el exterior de  $C$ , respectivamente. Entonces, se tiene que el funcional de energía estará formado por la suma del funcional interno ( $F_1$ ), es decir, la región que esté dentro del contorno de la curva  $C$ , y el funcional externo, es decir, la región comprendida fuera del contorno.

$$\begin{aligned} F_1(C) + F_2(C) &= \int_{\text{interno}(C)} |u_o(x, y) - c_1|^2 dx dy \\ &+ \int_{\text{externo}(C)} |u_o(x, y) - c_2|^2 dx dy \end{aligned} \quad (43)$$

Si la curva está fuera del objeto, entonces  $F_1(C) > 0$  y  $F_2(C) \approx 0$ ; del mismo modo si la curva está dentro del objeto,  $F_1(C) \approx 0$  pero  $F_2(C) > 0$ ; y por último si la curva  $C$  está dentro y fuera del objeto  $F_1(C) > 0$  y  $F_2(C) > 0$ . Para minimizar el funcional, la curva tiene que aproximarse al contorno, esto ocurre cuando  $F_1(C) \approx 0$  y  $F_2(C) \approx 0$ . En la Figura 8 puede contemplar estas situaciones y como se detectará el contorno minimizando  $F_1(C) + F_2(C)$ .

Como se ha visto, este funcional permite obtener una buena segmentación en casos simples, pero no se tiene control de la curva, por este motivo fue modificado añadiendo términos de regularización como la longitud y el área de la curva. De esta forma el funcional queda de la siguiente manera (ecuación 44),

$$\begin{aligned} F(c_1, c_2, C) &= \mu \cdot \text{long}(C) + \nu \cdot \text{área}(\text{interno}(C)) \\ &+ \lambda_1 \int_{\text{interno}(C)} (u_o(x, y) - c_1)^2 dx dy \\ &+ \lambda_2 \int_{\text{externo}(C)} (u_o(x, y) - c_1)^2 dx dy \end{aligned} \quad (44)$$

Donde  $c_1$  y  $c_2$  son constantes desconocidas, y  $\mu > 0$ ,  $v \geq 0$  y  $\lambda_1, \lambda_2 > 0$  son parámetros constantes pero dependientes de la imagen  $u_0$ .

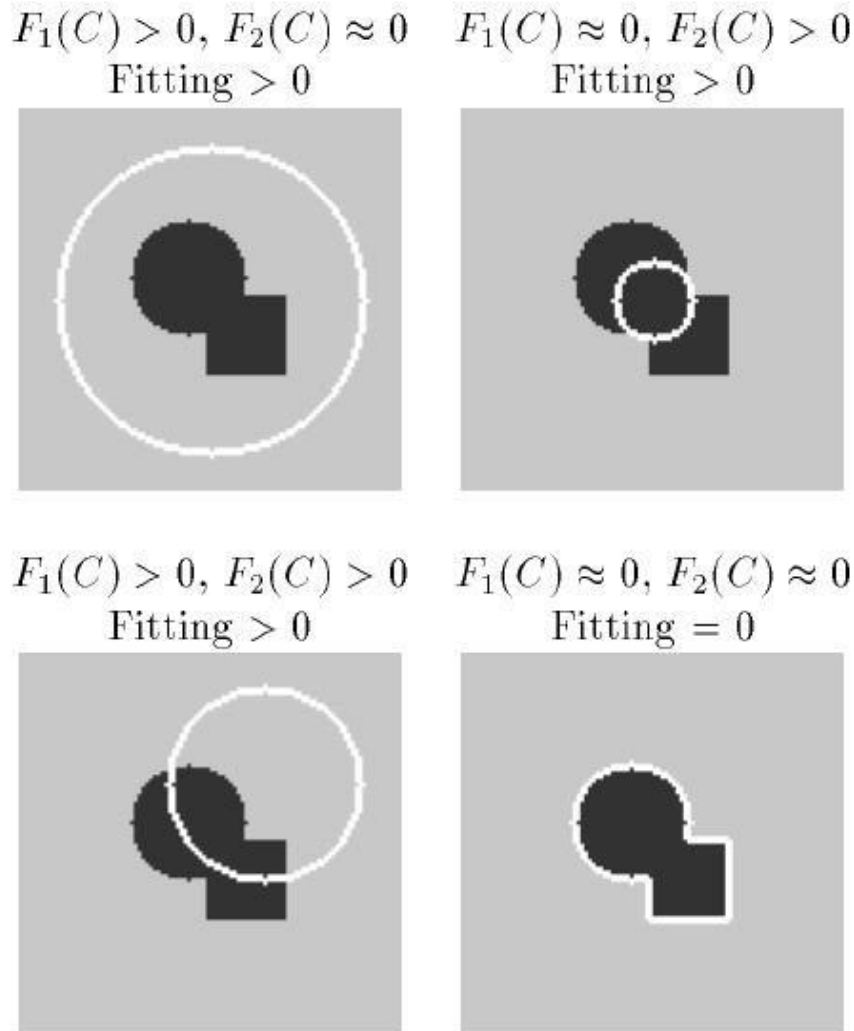


Figura 8. Casos posibles de localización de la curva  $C$ . La energía de minimización ("fitting energy") se minimiza cuando la curva está en el contorno del objeto (Chan & Vese, 2001)

#### 2.2.4. Relación entre los modelos Mumford-Shah y Chan-Vese

Escribiendo el funcional de Mumford-Shah con la nomenclatura usada en Chan Vese, se obtiene la ecuación 45,



$$F^{MS}(u, C) = \mu \cdot \text{long}(C) + \lambda \int_{\Omega} |u_0(x, y) - u(x, y)|^2 dx dy + \int_{\Omega/-c} |\nabla u(x, y)| dx dy \quad (45)$$

donde  $u_0$  es la imagen dada,  $\lambda$  y  $\mu$  parámetros positivos y  $u$  es la solución obtenida por minimización de este funcional.

Si en el funcional de Chan Vese se toma  $v = 0$ ,  $\lambda_1 = \lambda_2 = \lambda$ , se obtiene el de Mumford-Shah, por lo que se puede decir que el Chan Vese es un modelo general del Mumford-Shah (Chan & Vese, 2001).

#### 2.2.4.1. Formulación en *Level Sets*

Considerando  $C = \{ (x, y) \in \Omega : \varphi(x, y) = 0 \}$ , donde  $\varphi : \Omega \rightarrow \mathbb{R}$  es una función de Lipschitz. Por lo tanto, se puede definir,

$$\begin{aligned} \text{interno}(C) &= \omega = \{ (x, y) \in \Omega : \varphi(x, y) > 0 \}, \\ \text{externo}(C) &= \Omega \setminus \omega = \{ (x, y) \in \Omega : \varphi(x, y) < 0 \}, \end{aligned} \quad (46)$$

siendo  $\omega \subset \Omega$  abierto y por tanto  $C = \partial\omega$  (Chan y Vese, 2001).

El modelo Chan Vese puede reescribirse como un modelo de contornos activos reemplazando  $C$  por los conjuntos de nivel  $\varphi$ , de este modo el funcional se puede reescribir como en la ecuación 47,

$$\begin{aligned} F(\varphi, c_1, c_2) &= \mu \cdot \text{long}\{\varphi = 0\} + v \cdot \text{area}\{\varphi \geq 0\} \\ &+ \lambda_1 \int_{\varphi \geq 0} (u_o(x, y) - c_1)^2 dx dy \\ &+ \lambda_2 \int_{\varphi < 0} (u_o(x, y) - c_2)^2 dx dy \end{aligned} \quad (47)$$

Para reescribir el funcional mediante curvas de nivel, se va a usar, al igual que con Mumford-Shah, la función de Heaviside (ecuación 48) y también la función delta, que se corresponde a su derivada, y a la medida de Dirac en una dimensión (ecuación 49),



$$H(z) = \begin{cases} 1, & \text{si } z \geq 0, \\ 0, & \text{si } z < 0 \end{cases} \quad (48)$$

$$\partial(z) = \frac{d}{dt} H(z) \quad (49)$$

Utilizando estas funciones y se puede deducir las siguientes ecuaciones,

$$long\{\varphi = 0\} = \int_{\Omega} |\nabla H(\varphi)| = \int_{\Omega} \partial(\varphi) |\nabla \varphi(x, y)| dx dy, \quad (50)$$

$$area\{\varphi \geq 0\} = \int_{\Omega} H(\varphi(x, y)) dx dy, \quad (51)$$

$$\int_{\varphi \geq 0} (u_o(x, y) - c_1)^2 dx dy = \int_{\Omega} (u_o(x, y) - c_1)^2 H(\varphi(x, y)) dx dy, \quad (52)$$

$$\begin{aligned} \int_{\varphi \geq 0} (u_o(x, y) - c_2)^2 dx dy \\ = \int_{\Omega} (u_o(x, y) - c_2)^2 (1 - H(\varphi(x, y))) dx dy, \end{aligned} \quad (53)$$

De forma que sustituyendo en la ecuación 47, se obtiene,

$$\begin{aligned} J(\varphi, c_1, c_2) = & \mu \int_{\Omega} \partial(\varphi) |\nabla \varphi| + v \int_{\Omega} H(\varphi) dx \\ & + \lambda_1 \int_{\Omega} (u_o - c_1)^2 H(\varphi) dx \\ & + \lambda_2 (\varphi) dx \int_{\Omega} (u_o - c_2)^2 (1 - H(\varphi)) dx \end{aligned} \quad (54)$$

Donde los valores de  $c_1$  y  $c_2$  en función de  $\varphi$  son,

$$c_1 = \frac{\int_{\Omega} u_o H(\varphi) dx}{\int_{\Omega} H(\varphi) dx}, \quad (55)$$





$$c_2 = \frac{\int_{\Omega} u_o(1 - H(\varphi))dx}{\int_{\Omega} (1 - H(\varphi))dx},$$

Estas constantes  $c_1$  y  $c_2$  pueden interpretarse como una media de  $u_o$  en  $\{\varphi \geq 0\}$  y en  $\{\varphi < 0\}$ , respectivamente.

A partir de aquí, se debe minimizar la energía con respecto a  $\varphi$ , pero fijando  $c_1$  y  $c_2$ . Debido a que la función Heaviside es no diferenciable, se regulariza el problema cambiando  $H$  por  $H_\varepsilon$  de tal manera que sea posible calcular la derivada, de forma que el problema regularizado es,

$$\begin{aligned} J_\varepsilon(\varphi) = & \mu \int_{\Omega} \partial_\varepsilon(\varphi) |\nabla \varphi| + v \int_{\Omega} H_\varepsilon(\varphi) dx \\ & + \lambda_1 \int_{\Omega} (u_o - c_1)^2 H_\varepsilon(\varphi) dx \\ & + \lambda_2 \int_{\Omega} (u_o - c_2)^2 (1 - H_\varepsilon(\varphi)) dx \end{aligned} \quad (56)$$

Hay diversas formas de regularización, por ejemplo, la que proponen Chan y Vese es de dos tipos,  $H_1$  y  $H_2$ , que son regularizaciones  $C^2(\overline{\Omega})$  y  $C^\infty(\overline{\Omega})$ .

$$H_{1,\varepsilon}(z) = \begin{cases} 1, & \text{si } z > \varepsilon \\ \frac{1}{2} \left[ 1 + \frac{z}{\varepsilon} + \frac{1}{\pi} \sin\left(\frac{\pi z}{\varepsilon}\right) \right], & \text{si } |z| \leq \varepsilon \\ 0, & \text{si } z < -\varepsilon \end{cases} \quad (57)$$

$$\partial_{1,\varepsilon}(z) = \begin{cases} 0, & \text{si } |z| > \varepsilon \\ \frac{1}{2\varepsilon} \left[ 1 + \cos\left(\frac{\pi z}{\varepsilon}\right) \right], & \text{si } |z| \leq \varepsilon \end{cases} \quad (58)$$

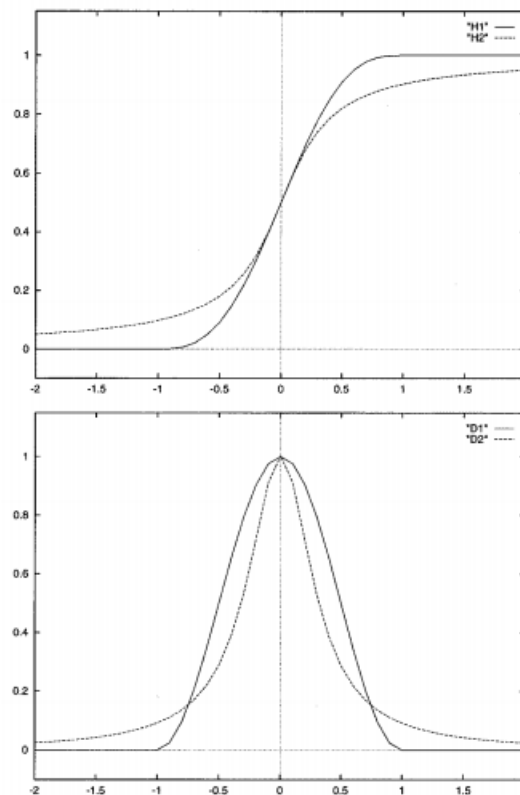
$$H_{2,\varepsilon}(z) = \frac{1}{2} \left( 1 + \frac{2}{\pi} \arctan\left(\frac{z}{\varepsilon}\right) \right) \quad (59)$$

$$\partial_{2,\varepsilon}(z) = \frac{\varepsilon}{\pi(\varepsilon^2 + z^2)} \quad (60)$$

Ambos tienden a  $H$  y a  $\delta$  cuando  $\varepsilon \rightarrow 0$ . La diferencia se basa en que el intervalo de  $\partial_1$  es  $[-\varepsilon, \varepsilon]$  y el intervalo  $\partial_2$  no es nulo en todo el dominio.



En la Figura 9 se puede observar las regularizaciones en la función Heaviside y en la medida de Dirac.



*Figura 9. Diferentes regularizaciones de la función Heaviside(arriba) y la función delta(abajo) (Chan y Vese, 2001)*

### 2.2.5. Distancia de regularización de la evolución del Level Set (DRLSE)

Los métodos *Level Set* geométricos han sido muy utilizados en la segmentación y procesamiento de imágenes, ya que presenta ventajas como que pueden representar contornos complejos y se adaptan a cambios de topología complejas, pero presenta irregularidades que puede causar errores numéricos y provocar una evolución inestable. Para solucionarlo se utiliza la reinicialización, de forma que la función de ajuste de nivel sea reemplazada periódicamente y degradada con una función de signo, pero afecta a la precisión numérica además de que se desconoce el cómo y cuándo se hace (Li, Xu, Gui, & Fox, 2010).



C.Li, C.Xu, C.Gui y M.D.Fox proponen una nueva formulación de *Level Set* variacionales donde se mantiene la regularidad de la función de ajuste de niveles durante su evolución. La evolución del *Level Set* se deriva como el flujo de gradiente con un término de regularización que minimiza una energía funcional y una energía externa que impulsa el movimiento del *Level Set* cero a la ubicación deseada. La idea básica de la propuesta de los autores es representar *Level Set* cero y formular su movimiento.

El término de regularización de distancia se define como una función tal que la derivada de los *Level Set* tienen un efecto de difusión con un avance y retroceso único siendo capaz de mantener la forma deseada de la curva de la función de ajuste de nivel cerca del *Level Set* cero. Esto produce un nuevo tipo de evolución de *Level Set* llamado distancia de regularización. La distancia de regularización elimina la necesidad de reinicialización, y, por tanto, evita errores numéricos y se denota como DRSLE.

### 2.2.5.1. Descripción del método

Siendo el contorno paramétrico dinámico  $C(s,t) : [0,1] \times [0,\infty] \rightarrow \mathbb{R}^2$  la evolución de la curva puede expresarse como:

$$\frac{\partial C(s,t)}{\partial t} = FN \quad (61)$$

Donde  $F$  es la función velocidad, que controla el movimiento del contorno,  $N$  es el vector normal hacia el interior de la curva y  $C(s,t)$  es el *Level Set* cero dependiente de  $\varphi(x,y,t)$ . Suponiendo que la función de ajuste de nivel toma valores negativos en el interior del contorno del *Level Set* cero y valores positivos en el exterior, el vector normal hacia el interior se puede expresar como  $-\frac{\nabla\varphi}{|\nabla\varphi|}$ , donde  $\nabla$  es el operador gradiente. La evolución de la curva puede expresarse entonces como,

$$\frac{\partial\varphi}{\partial t} = F|\nabla\varphi| \quad (62)$$



Como se ha mencionado anteriormente, la función de ajuste de nivel presenta irregularidades en su evolución que puede llevar a errores numéricos e inestabilidad. Para restaurar la estabilidad y mantener la evolución del *Level Set* se utiliza la reinicialización, que se basa en detener periódicamente la evolución y remodelar la LSF como una función de distancia. Una forma de reinicialización es,

$$\frac{\partial \Psi}{\partial t} = \text{sign}(\Psi)(1 - |\nabla \varphi|) \quad (63)$$

Donde  $\varphi$  es la función de ajuste de nivel para la reinicialización y  $\text{sign}(\cdot)$  es el signo de la función. Aunque la reinicialización mantiene la regularidad, la función de ajuste de nivel puede desviarse del nivel cero, por lo tanto, se debe evitar la reinicialización tanto como sea posible (Li, Xu, Gui, & Fox, 2010).

#### 2.2.5.2. Formulación de la energía con la regularización de distancia

Para  $\Phi: \Omega \rightarrow \mathbb{R}$  siendo  $\Omega$  el dominio de la función *Level Set*, la energía funcional se define como,

$$\varepsilon(\varphi) = \mu R_p(\varphi) + \varepsilon_{ext}(\varphi) \quad (64)$$

Donde  $R_p(\varphi)$  es el término de regularización,  $\mu > 0$  es una constante y  $\varepsilon_{ext}(\varphi)$  es la energía exterior que depende de los datos de interés, esta energía es mínima cuando el *Level Set* cero de la función *Level Set* está localizada en la región deseada. El término de regularización del *Level Set* se define como,

$$R_p(\varphi) \triangleq \int_{\Omega} p |\nabla \varphi| dx \quad (65)$$

Donde  $p$  es una función potencial  $p: [0, \infty) \rightarrow \mathbb{R}$ . Este potencial se puede definir de forma sencilla como



$$p = p_1(s) \triangleq \frac{1}{2}(s - 1)^2 \quad (66)$$

Donde  $s=1$  es el único punto mínimo. Con este potencial  $p=p_1(s)$ , el término de regularización del *Level Set* puede expresarse como,

$$P(\varphi) = \frac{1}{2} \int_{\Omega} (|\nabla \varphi| - 1)^2 dx \quad (67)$$

Que caracteriza la desviación de  $\varphi$  con una función de distancia de signo. Sin embargo, la evolución del *Level Set* para la energía de minimización tiene efectos no deseados en la LSF en algunas circunstancias, por ejemplo, cuando se tiene  $|\nabla \varphi| = 0$  en el interior del contorno la difusión aumenta demasiado y produce una oscilación en  $\varphi$ , lo cual se muestra como picos y valles en la curva que define el contorno y la LSF. Para evitar esto, se añade una nueva función potencial en el término de distancia de regularización  $R_p$ . Esta nueva función potencial mantiene la propiedad  $|\nabla \varphi| = 1$  en los píxeles vecinos del *Level Set* cero, mientras que se mantiene LSF como una constante, con  $|\nabla \varphi| = 0$  en localizaciones lejanas al *Level Set* cero. Para mantener esto, la función potencial debe tener dos puntos mínimos, en  $s=1$  y  $s=0$ , de forma que soluciona los problemas que dejaba la ecuación 67 y además ofrece otras propiedades en la evolución del *Level Set*, siendo la nueva función potencial  $p_2(s)$ ,

$$p_2(s) = \begin{cases} \frac{1}{4\pi^2} (1 - \cos(2\pi s)), & si \ s \leq 1 \\ \frac{1}{2} (s - 1)^2, & si \ s \geq 1 \end{cases} \quad (68)$$

### 2.2.5.3. Aplicación a la segmentación de imágenes

La formulación de distancia del *Level Set* puede tener muchas aplicaciones con distintas definiciones de energía externa  $\varepsilon_{ext}$ . Para aplicaciones de segmentación, una variedad de la información de la imagen, incluida la formación de las regiones o la de los bordes, se puede usar para definir esta energía.



#### 2.2.5.4. Formulación de distancia de regularización de *Level Set* basado en bordes de contornos activos

Siendo  $I$  una imagen en el dominio  $\Omega$ , se define un indicador de bordes como una función  $g$ ,

$$g \triangleq \frac{1}{1 + |\nabla G_\sigma * I|^2} \quad (69)$$

Donde  $G_\sigma$  es una función Gaussiana con una desviación estándar  $\sigma$ . La convolución con la imagen se usa para suavizarla y reducir el ruido. Esta función  $g$  normalmente toma valores mejores en los bordes del objeto que en otras localizaciones.

Para una LSF  $\Phi: \Omega \rightarrow \mathbb{R}$ , se define la energía funcional como,

$$\varepsilon(\varphi) = \mu R_p(\varphi) + \lambda L_g(\varphi) + \alpha A_g(\varphi) \quad (70)$$

Donde  $\lambda > 0$ ,  $\alpha \in \mathbb{R}$  son coeficientes de los funcionales de energía. El funcional de energía  $L_g(\varphi)$ , correspondiente con la longitud ponderada de la curva, queda definido como,

$$L_g(\varphi) \triangleq \int_{\Omega} g \delta(\varphi) |\nabla \varphi| dx \quad (71)$$

Donde  $\delta$  es la función delta de Dirac, con la que  $L_g$  se define como una integral de línea de la función  $g$  a lo largo del contorno de *Level Set* de  $\varphi$ . Cabe destacar que se propuso un modelo de contornos activos geodésicos parametrizando el nivel cero de  $\varphi$  como un contorno  $C[0,1] \rightarrow \Omega$ , en el que la energía  $L_g(\varphi)$  puede expresarse como en la ecuación 47 (Caselles, Kimmel, & Sapiro, 1997),

$$L_g(\varphi) \triangleq \int_0^1 g(C(s)) |C'(s)| ds \quad (72)$$

El último funcional de energía  $A_g(\varphi)$  de la ecuación 70 se define como,



$$A_g(\varphi) \triangleq \int_{\Omega} gH(-\varphi)dx \quad (73)$$

Donde  $H$  es la función Heaviside. La función  $A_g(\varphi)$  engloba un área ponderada de la región  $\Omega_{\varphi}^{-} \triangleq \{x: \varphi(x) < 0\}$ . En el caso de  $g=1$ , el área es exactamente la región  $\Omega_{\varphi}^{-}$ . La energía  $A_g(\varphi)$  acelera el movimiento del contorno nivel cero en el proceso de evolución establecido, lo cual es necesario cuando el contorno inicial se sitúa lejano a los bordes deseados.

Tomando los valores negativos dentro del objeto y los valores positivos fuera del mismo, si el contorno inicial está situado fuera del objeto, el coeficiente  $\alpha$  debe ser positivo, así que el contorno nivel cero puede reducir la evolución del *Level/ Set* y, de la misma forma, si el contorno inicial está localizado dentro del objeto,  $\alpha$  es negativo, por lo que el contorno tiende a expandirse.

A partir de esto, se puede decir que la función de  $g$  en  $A_g$  es ralentizar la reducción o ampliación del *Level/ Set* cero cuando llega al borde del objeto, tomando  $g$  valores muy pequeños.

En la práctica de los métodos de *Level/ Set*, la función Dirac,  $\delta_{\varepsilon}(x)$ , (ecuación 74) y Heaviside,  $H_{\varepsilon}(x)$ , (ecuación 75) se aproximan mediante estas funciones suavizadas, sabiendo que la función  $\delta_{\varepsilon}$  es la derivada de la  $H_{\varepsilon}$ , y el parámetro  $\varepsilon$  normalmente se fija con el valor 1.5.

$$\delta_{\varepsilon}(x) = \begin{cases} \frac{1}{2\varepsilon} [1 + \cos(\frac{\pi x}{\varepsilon})], & |x| \leq \varepsilon \\ 0, & |x| > \varepsilon \end{cases} \quad (74)$$

$$H_{\varepsilon}(x) = \begin{cases} \frac{1}{2} [1 + \frac{x}{\varepsilon} + \frac{1}{\pi} \sin(\frac{\pi x}{\varepsilon})], & |x| \leq \varepsilon \\ 1, & x > \varepsilon \\ 0, & x < -\varepsilon \end{cases} \quad (75)$$

Sustituyendo estas funciones en la ecuación 70 se obtiene,



$$\varepsilon(\varphi) = \mu \int_{\Omega} p(|\nabla \varphi|) d\mathbf{x} + \lambda \int_{\Omega} g \delta_{\varepsilon}(\varphi) |\nabla \varphi| d\mathbf{x} + \alpha \int_{\Omega} g H_{\varepsilon}(-\varphi) d\mathbf{x} \quad (76)$$

Esta energía funcional puede minimizarse resolviendo el siguiente flujo de gradiente cuya ecuación es una aplicación general de la formulación DRSLE (ecuación 77).

$$\frac{\partial \varphi}{\partial t} = \mu \operatorname{div}(d_p(|\nabla \varphi|) |\nabla \varphi|) + \lambda \delta_{\varepsilon}(\varphi) \operatorname{div}\left(g \frac{\nabla \varphi}{|\nabla \varphi|}\right) + \alpha g \delta_{\varepsilon}(\varphi) \quad (77)$$

Dada una función inicial LSF  $\varphi(x, 0) = \varphi_o(x)$ . El primer término de la ecuación anterior  $\frac{\partial \varphi}{\partial t}$ , está asociado con la distancia de regularización  $R_p(\varphi)$ , y el segundo y tercer término están relacionados con  $L_g(\varphi)$  y  $A_g(\varphi)$ , respectivamente.

#### 2.2.5.5. Implementación en banda estrecha

El coste computacional del método *Level Set* puede reducirse a partir de la implementación en banda estrecha del *Level Set* cero. La aplicación de banda estrecha requiere reinicialización más frecuente, por lo que su aplicación es más sofisticada. La implementación de la banda estrecha en DRSLE consiste en la actualización de la LSF para reducir el número de iteraciones y el tiempo de cómputo.

Siendo  $\varphi(i, j)$  una LSF definida en una cuadrícula, e  $(i, j)$  un punto de la misma, si  $\varphi_{i-1, j}$  y  $\varphi_{i+1, j}$ , o  $\varphi_{i, j-1}$  y  $\varphi_{i, j+1}$  tienen signos contrarios, se le llama punto cero de cruce. El conjunto de todos los puntos cero de cruce del LSF se denotan como  $Z$ . Luego la banda estrecha se define como,

$$B_r = \bigcup_{(i, j) \in Z} N_{i, j}^{(r)} \quad (78)$$

Donde  $N_{i, j}^{(r)}$  es un bloque cuadrado centrado en el punto  $(i, j)$ . Se fija  $r$  con un valor mínimo de  $r=1$ , en cuyo caso la banda estrecha  $B_r$  es la unión 3x3 de los puntos cero de cruce vecinos.





La implementación del método de banda estrecha en DRSLE consiste en los siguientes pasos:

1. **Inicialización.** Se inicia  $\varphi$  como una función  $\varphi_0$ . Entonces, la banda estrecha inicial es  $B_r^0$  (ecuación 79),

$$B_r^0 = \bigcup_{(i,j) \in Z^0} N_{i,j}^{(r)} \quad (79)$$

donde  $Z^0$  es el conjunto de puntos cero de cruce de  $\varphi_0$ .

2. **Actualizar el LSF.** Actualizar  $\varphi_{i,j}^{k+1} = \varphi_{i,j}^k + \tau L(\varphi_{i,j}^k)$  en la banda estrecha  $B_r^k$ .
3. **Actualizar la banda estrecha.** Determinar el conjunto de todos los puntos cero de cruce de  $\varphi_{i,j}^{k+1}$  en  $B_r^k$  se denota como  $Z^{k+1}$ . Entonces, se actualiza la banda estrecha como,

$$B_r^k = \bigcup_{(i,j) \in Z^{k+1}} N_{i,j}^{(r)} \quad (80)$$

4. **Asignar valores a los píxeles nuevos de la banda estrecha.** Para cada punto  $(i,j)$  en  $B_r^{k+1}$  pero no en  $B_r^k$ , se asigna  $\varphi_{i,j}^{k+1}$  a  $h$ , si  $\varphi_{i,j}^k > 0$ , o si no se asigna  $\varphi_{i,j}^{k+1}$  a  $-h$ , siendo  $h$  una constante, que puede fijarse como un valor genérico  $r+1$ .
5. **Determinar el final de la iteración.** Si cualquiera de los puntos cero de cruce dejan de variar en  $m$  iteraciones consecutivas o  $k$  excede un número máximo de iteraciones, entonces se para la iteración, en otro caso, se retoma el paso 2 (Li, Xu, Gui, & Fox, 2010).





## 3. Desarrollo y análisis de resultados

### 3.1. DESCRIPCIÓN DEL PROBLEMA

Se cuenta con unas imágenes de casos clínicos reales proporcionadas por un profesional sanitario de forma que incluye imágenes construidas por la máquina de resonancia magnética en formato DICOM e imágenes con ROI, es decir, imágenes donde se ha marcado ya la zona de interés, en este caso, un tumor cerebral.

Se tomará la imagen con ROI como la imagen de referencia y se intenta que la curva del *snake* se ajuste lo máximo posible a este contorno. Para ello, se utiliza la base teórica explicada anteriormente y su implementación en MATLAB.

Posteriormente, se realiza una comparación entre el contorno obtenido y el original hallando la curva ROC para comprobar la fiabilidad de este método.

Por último, se realiza un desarrollo sistemático del parámetro alpha, que influye sobre la curva del *snake* significativamente, de modo que se representa la evolución de la curva dependiendo del valor de este parámetro para posteriormente elegir el valor óptimo de alpha en estos casos.

### 3.2. IMPLEMENTACIÓN DEL MÉTODO

En este trabajo se ha desarrollado la implementación del modelo de contornos activos basados en bordes como una aplicación del método basado en DRSLE (Li, Xu, Gui, & Fox, 2010).

La implementación de este método en Matlab ha sido realizada por Chuning Li, como una función cuyas variables son:

- $\phi_0$ : función de *Level Set* (LSF) que se actualizará según la evolución del *Level Set*.



g: función de indicador de borde. Expande o contrae el contorno cuando se ha encontrado el contorno objetivo. Esta función está inversamente relacionada con la convolución gaussiana de la imagen, la cual tiene una desviación estándar que junto con el tamaño que se quiera definir el filtro gaussiano, es lo que influye en el resultado del *snake*.

El parámetro sigma representa la desviación estándar y tiene que ser mayor que cero. En la Figura 10 se puede ver como se suaviza el *snake* haciendo la función de filtrado gaussiano, siendo mayor el suavizado cuanto mayor sea sigma. Al tener un valor alto de sigma disminuye la sensibilidad del *snake* para captar la forma del tumor, dando como resultado la estructura principal, esto es útil si se quiere que la curva no evolucione a zonas que no son de interés, pero no lo es si se necesita segmentar estructuras pequeñas o con cambios bruscos de intensidad. En estos casos se debe fijar un valor de sigma bajo.

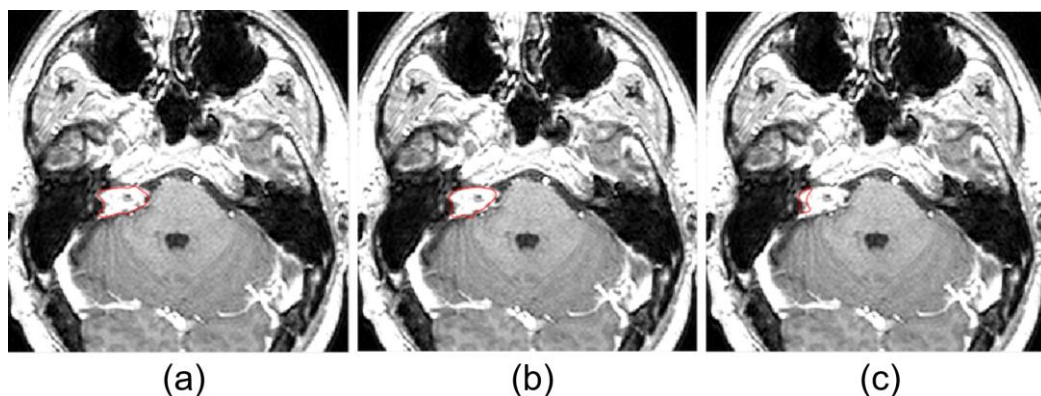


Figura 10. a)  $\sigma=0.5$  b)  $\sigma=1.8$  c)  $\sigma=5$

Por otro lado, el parámetro hsize es el tamaño del filtro gaussiano, que como se puede ver en la Figura 11, su influencia en el resultado es mínima.

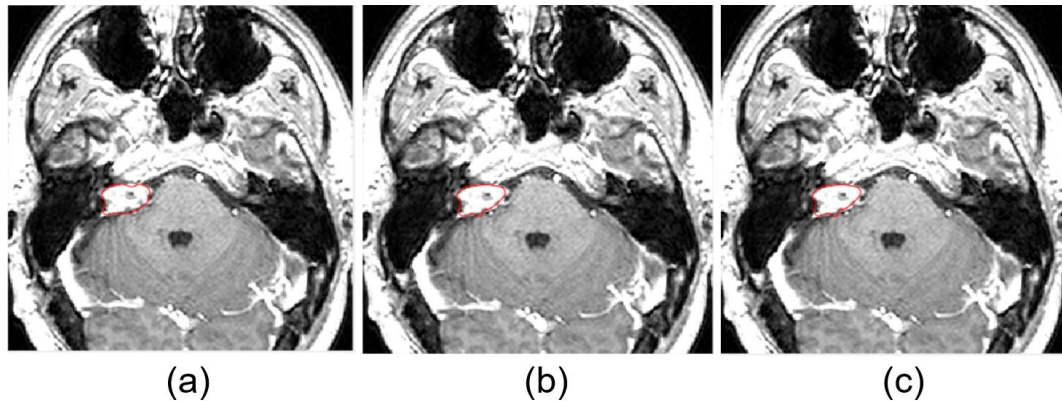


Figura 11. a)  $hsize=2$  b)  $hsize=20$  c)  $hsize=50$

- timestep: parámetro de paso de tiempo  $\Delta t$  al correr el programa.

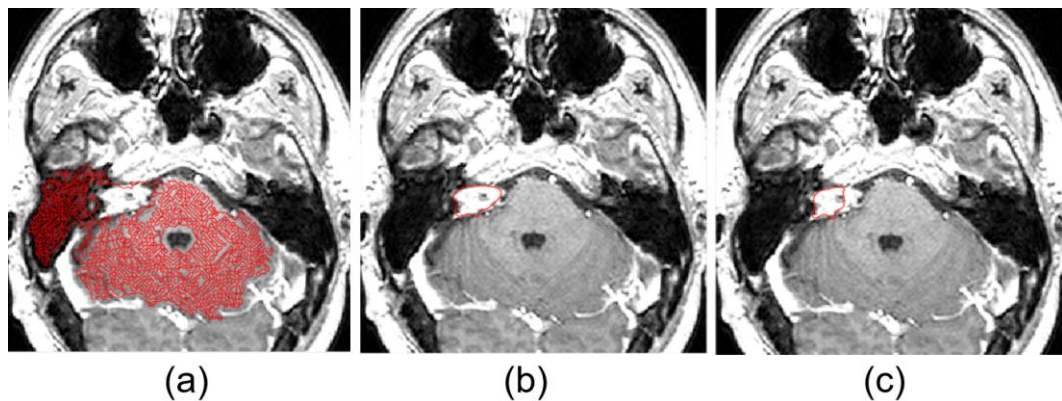


Figura 12. a)  $timestep=-2$  b)  $timestep=1$  c)  $timestep=5$

Tal y como se puede ver en la Figura 12, el timestep no puede ser negativo, el contorno se vuelve muy irregular, ya que por lógica, no se puede representar un tiempo negativo. Cuanto mayor sea el timestep, más se ajustará la curva al objeto que se desea segmentar, pero con unos valores demasiado altos el snake evoluciona de manera no deseada, introduciéndose dentro del contorno objetivo. En los casos representados en la Figura 12, un  $timestep=1$  ajusta la curva de la forma deseada, pero si se aumenta a un valor de 5 el contorno se contrae más de lo esperado.

- $\mu$ : peso del término de regularización de distancia, es decir, se encarga del suavizado de la función  $\phi$ , y también cumplir en al menos una zona vecina la “distancia con signo”  $|\nabla\phi| = 1$ , para que la evolución de la curva sea más exacta. Este parámetro se define como  $0.2/timestep$ .





- $\lambda$ : parámetro que define el peso del término de energía de longitud ponderada  $L_g$ . Esta energía se minimiza cuando el contorno de la función  $\phi$  se localiza en los bordes del objeto que se quiere segmentar.

En la Figura 13 se puede observar como el parámetro  $\lambda$  no influye demasiado, para que la curva se deforme se debe poner valores muy altos. Este parámetro debe ser mayor que cero.

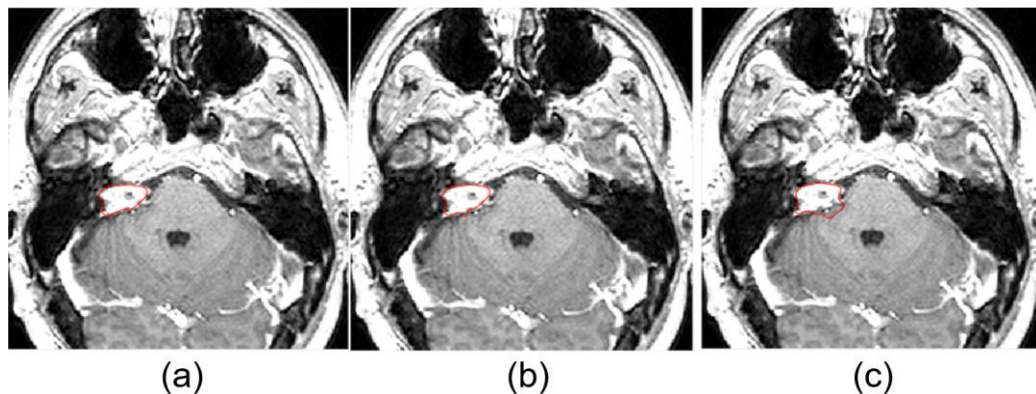


Figura 13. a)  $\lambda=1$  b)  $\lambda=10$  c)  $\lambda=50$

- $\alpha$ : peso del término del área ponderada. Esta función sirve para acelerar el movimiento del contorno en el proceso de evolución cuando la curva inicial está lejos del contorno deseado. Se define positivo cuando el contorno inicial está fuera del objeto y negativo cuando está dentro del objeto. Para refinar el contorno, se fija  $\alpha=0$ , de forma que este término no influya en los resultados finales.

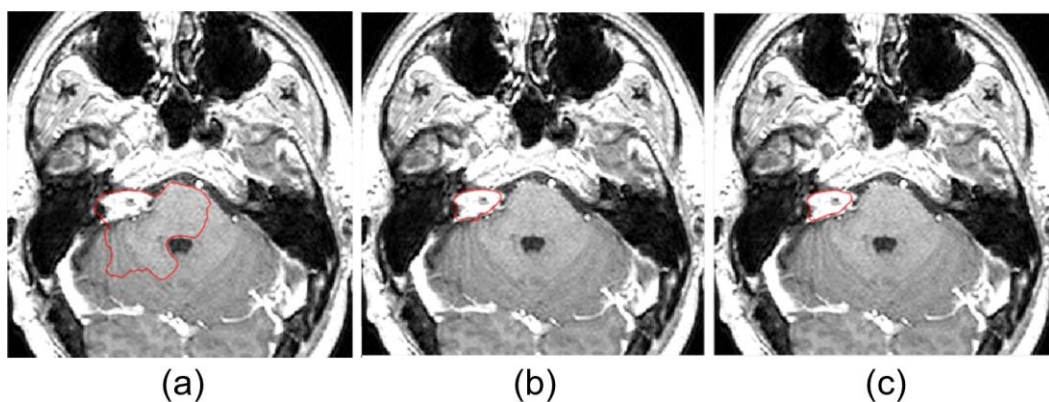


Figura 14. a)  $\alpha=-5$  b)  $\alpha=5$  c)  $\alpha=10$



Alpha es el parámetro que más influye a la hora de ajustar el contorno a los bordes deseados. La influencia de este parámetro será estudiada más adelante con más profundidad.

- epsilon: ancho de la función Delta de Dirac, la cual es la integral de línea de Lg. Este parámetro, si no se escoge debidamente, varía la curva bastante. Es recomendable fijar este parámetro con un valor alrededor de 1.

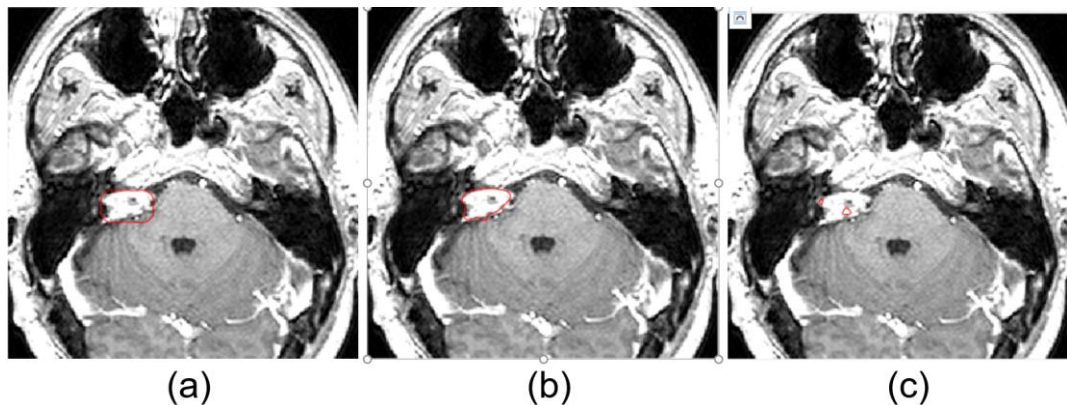


Figura 15. a)  $\epsilon=-5$  b)  $\epsilon=1$  c)  $\epsilon=5$

- iter: número de iteraciones del programa. Hay dos iteraciones por definir: iter\_inner e iter\_outer, que son respectivamente, el número de iteraciones dentro y fuera del contorno.

Como se puede observar en las Figuras 16 y 17, el número de iteraciones refina el contorno del *snake*, pero para que se ajuste lo máximo posible y no se contraiga demasiado se tiene que definir un valor lo más certero de este parámetro. Si el número de iteraciones es demasiado bajo, el contorno no evoluciona de la manera deseada y deja la segmentación incompleta. El número de iteraciones debe restringirse una vez se ha encontrado la estructura deseada ya que con altas iteraciones puede llegar a colapsar el *snake*.

Las iteraciones finales que realizará el programa son: iter\_inner · iter\_outer + iter\_refine. Estas últimas iteraciones son las de refinamiento del contorno, las cuales se han realizado con  $\alpha=0$ , y se fijan de forma que iter\_refine=10.

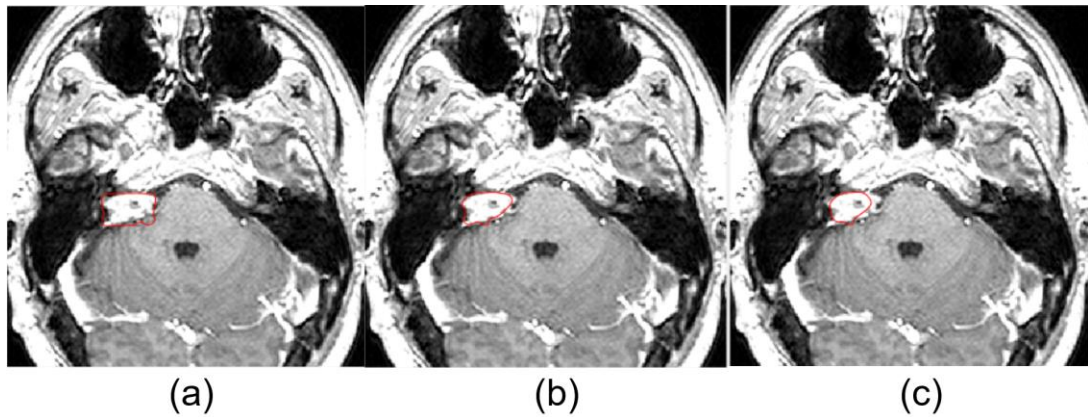


Figura 16. (a)  $Iter\_inner=2$  (b)  $Iter\_inner=40$  (c)  $Iter\_inner=100$

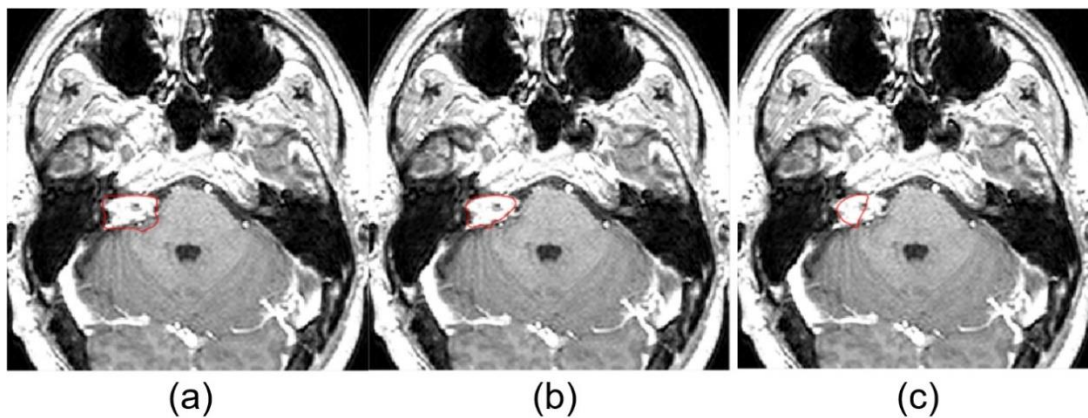


Figura 17. (a)  $Iter\_outer=0$  (b)  $Iter\_outer=10$  (c)  $Iter\_outer=50$

- **potentialFunction**: elección de la función potencial en el término de distancia de regularización. Se proporcionan dos opciones: **potentialFunction = 'single-well'** o **potentialFunction = 'double-well'**, que corresponde a las funciones potenciales  $p_1$  (potencial único) y  $p_2$  (doble potencial), respectivamente.
- **phi**: función de Level Set actualizada después de la evolución del Level Set. Es la curva que se marca inicialmente y que va a deformarse, siendo la única variable de salida del programa.

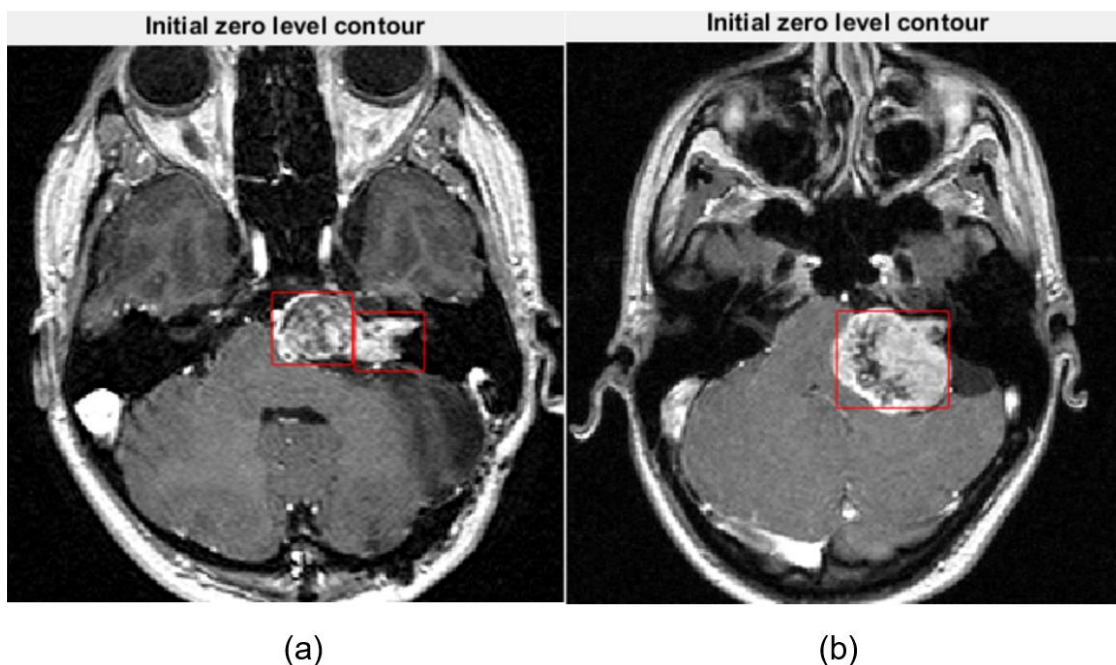
Para poder llevar a cabo un ajuste sensible de la curva *Level Set* cero al contorno deseado, primero ha de definirse una región que se sitúe aproximadamente en la misma localización del tumor y definir un rectángulo de área aproximada a la del tumor. Para ello, se debe especificar en el programa las coordenadas  $x$  e  $y$





de los vértices del rectángulo, es decir, desde donde hasta donde se desea que se dibuje el mismo, tanto en el eje x como el eje y.

Si la zona a estudiar es muy heterogénea, se puede definir más de un rectángulo inicial de forma que el área que engloben dichos rectángulos se ajuste más al contorno que se quiere segmentar. Si estas regiones se sitúan cerca una de las otras, por las características de evolución *Level Set* van a fusionarse para dibujar el contorno deseado. En la Figura 18 (a), se muestra un ejemplo con dos *Level Set* iniciales y en la Figura 20 (a), se muestra cómo evoluciona la curva obteniendo un *Level Set* final ajustado al contorno donde se han fusionado los dos *Level Set* iniciales.



*Figura 18. Dibujo del Level Set cero alrededor del tumor. a) Level Set cero inicial doble b) Level Set cero inicial único.*

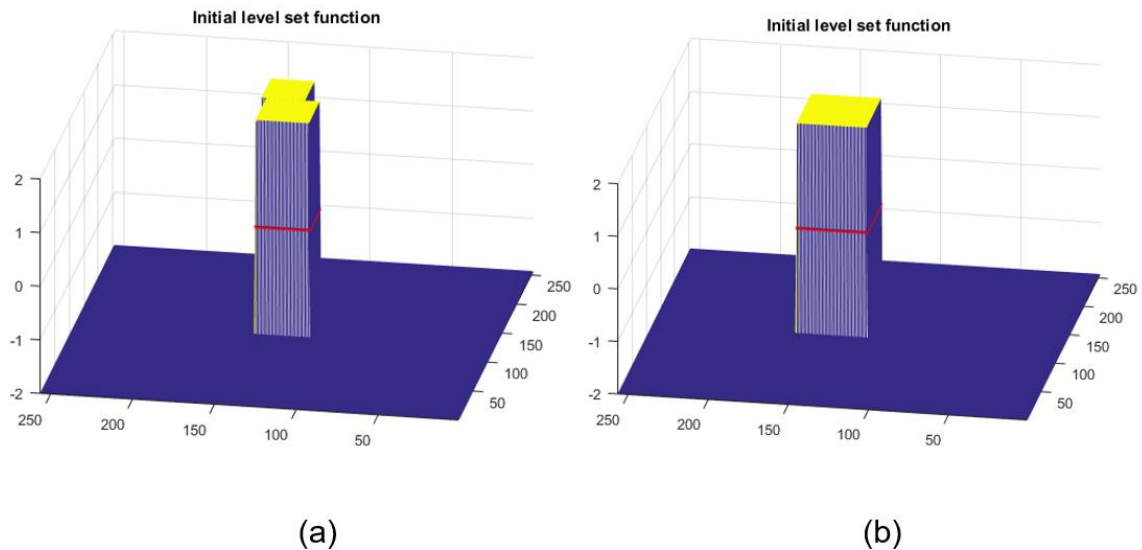
A continuación, para poder mostrar mejor el *Level Set*, se representa una gráfica en la que se muestra la zona seleccionada en la imagen, el plano de la base representa la imagen, con longitud 256x256, que es el tamaño de la imagen en píxeles y la tercera dimensión, la cual se fija una altura máxima 2, en la se puede ver la forma que se ha escogido como *Level Set* cero inicial.

Una vez el procedimiento se ha llevado a cabo, comparando la Figura 18 y la Figura 20 se puede observar cómo actúa el método DRSLE, ajustando la función

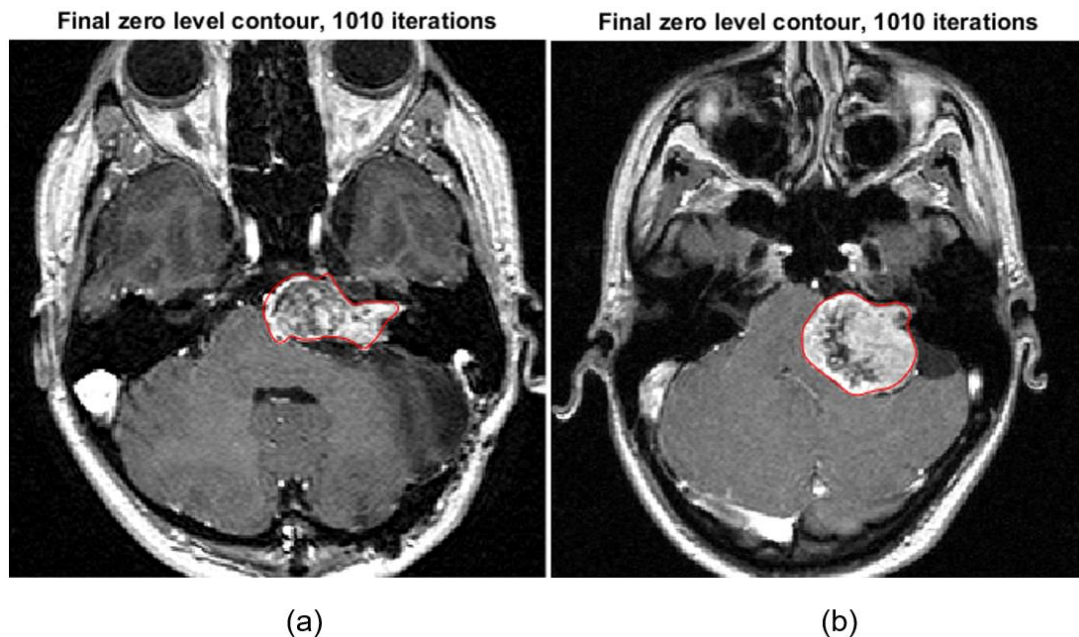


*Level Set* al contorno deseado. A partir de un rectángulo inicial marcado alrededor del tumor, la curva se ha adaptado al contorno de este.

Para poder llegar a esta solución el programa debe pasar una serie de iteraciones, que, como ya se ha explicado anteriormente es la multiplicación de las iteraciones de entrada por las de salida sumadas a las iteraciones de refinamiento.



*Figura 19. Representación de la función Level Set inicial*



*Figura 20. Contorno final adaptado al objeto específico*



Se dibuja de nuevo una gráfica que muestra de nuevo, en un plano de 256x256 que representa la imagen, un relieve que se corresponde con el contorno segmentado y ajustado por el *snake*.

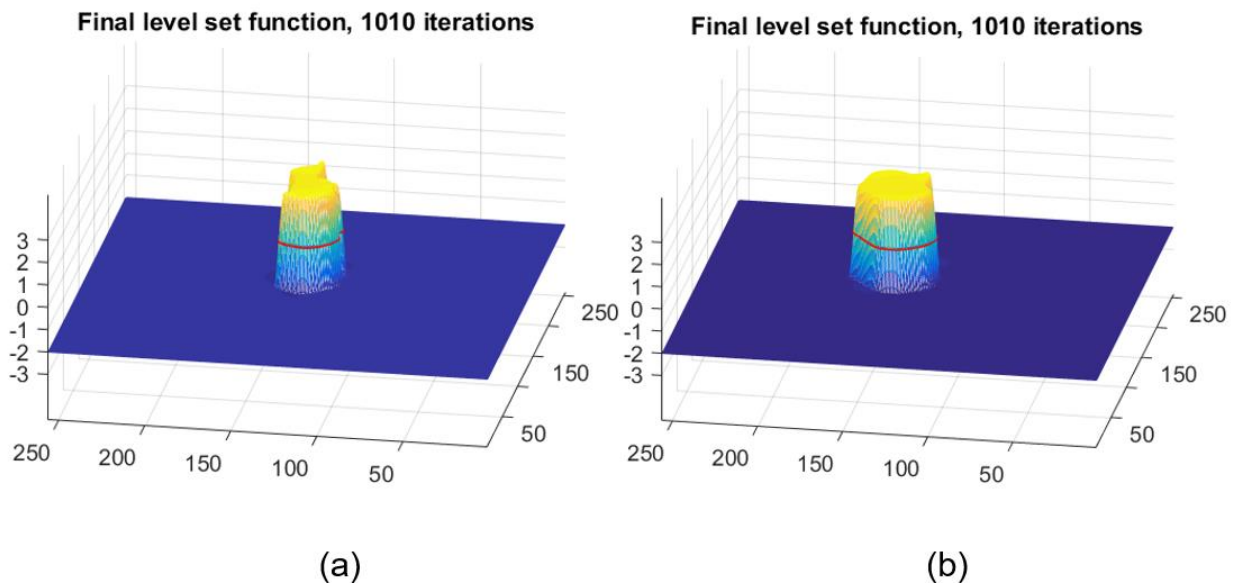
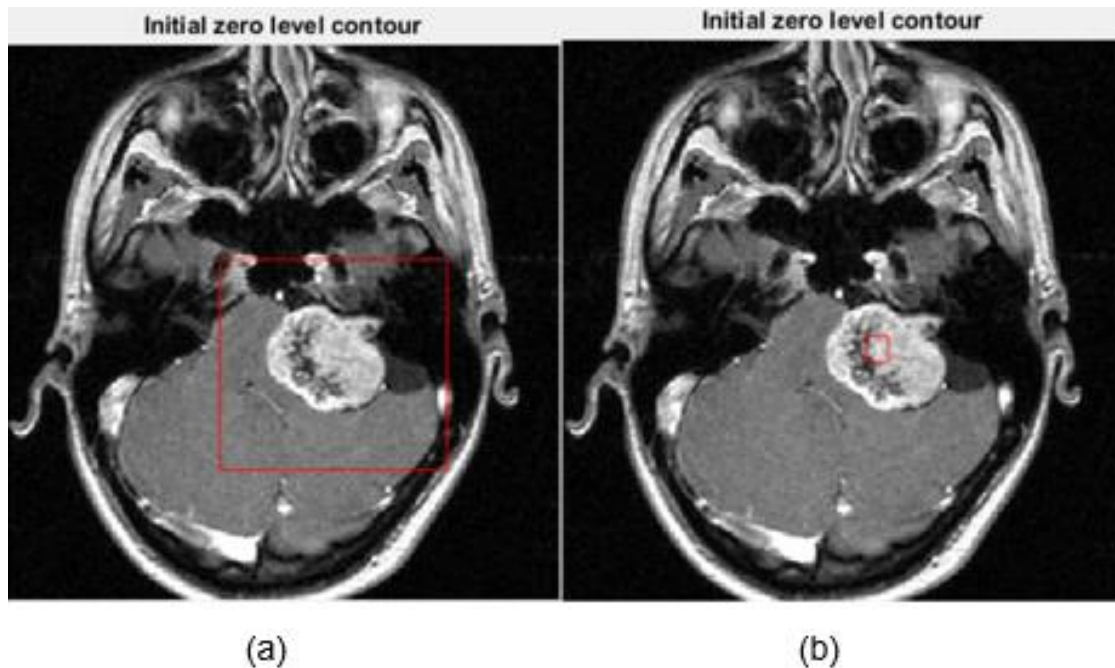


Figura 21. Representación de la función Level Set Final

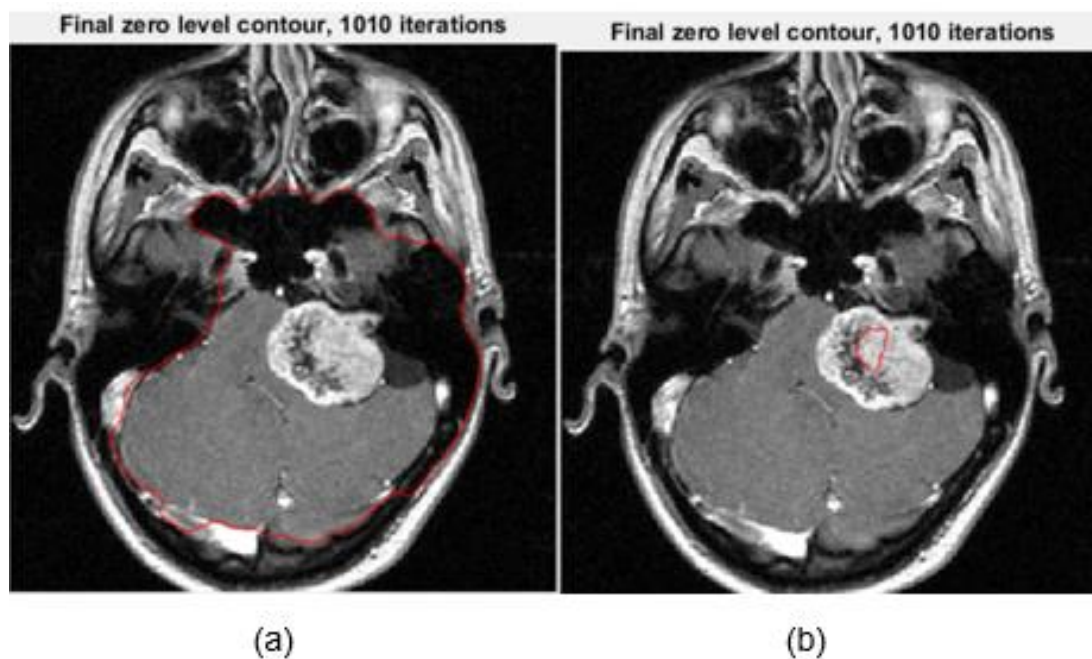
Los ejemplos representados en las Figuras 18 y 20 se han realizado con *Level Set* inicial ajustado a la zona del tumor, sin embargo, los resultados del *snake* final varían significativamente si se toma un *Level Set* inicial que no se ajuste al contorno.

En la Figura 22 se muestran dos casos, el primero en el que se sitúa el *Level Set* inicial en una región que comprende el tumor, pero es el área es demasiado amplia, y en el segundo caso una región dentro del tumor, pero demasiado pequeña.

En la Figura 23 se observa cómo evoluciona la curva en estos dos casos, resultando un *snake* que no se ajusta al contorno deseado.



*Figura 22. Level Set iniciales no ajustados*



*Figura 23. Level Set finales no ajustados*

Las conclusiones a las que se pueden llegar al realizar este estudio es que el resultado de la segmentación se ve altamente influenciado por cómo se haya definido el contorno inicial y, cuanto más ajustado sea el contorno al tumor a estudiar, menor será la probabilidad de que la curva evolucione a estructuras erróneas.





### 3.2.1. Estudio del parámetro alpha.

Para poder estudiar el comportamiento del parámetro alpha, se ha realizado un programa que va variando el valor de alpha en un rango de -10 hasta 10 y muestra los resultados del *snake* por cada valor de alpha. Como el resto de los parámetros son menos influyentes, se han determinado unos valores fijos de los mismos, siendo

`iter_inner=50;`

`iter_outer=20;`

`lambda=5;`

`epsilon=1.5;`

`sigma=1.8;`

y el tamaño de la función gaussiana  $H=20$ .

Para poder hacer un estudio equitativo, se fija un intervalo de iteraciones de alpha aumentando su valor de uno en uno, de forma que vaya mostrando la solución para cada valor.

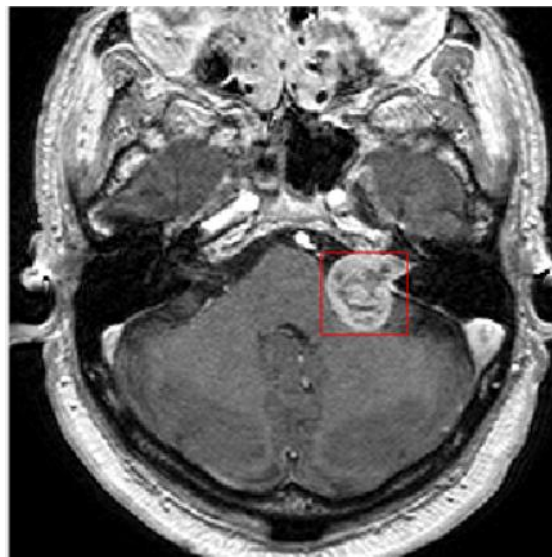
Al realizar este desarrollo sistemático se puede ver como en valores de alpha excesivamente pequeños, la curva dibuja un contorno demasiado grande que no se ajusta en ningún caso al *Level Set* inicial que se había dibujado, pero conforme alpha aumenta su valor, la curva va evolucionando rápidamente hasta que se ajusta a los bordes de la región de interés. Una vez la curva del *snake* está encima de los bordes de interés la evolución de la curva se ralentiza, y varía mínimamente, hasta que llega un punto que el valor de alpha es demasiado grande y la curva se desestabiliza.

Un alpha distinto de cero proporciona una fuerza externa adicional de forma que evita el movimiento del contorno, pero el contorno final puede desviarse ligeramente de la verdadera frontera del objeto debido al efecto de la contracción o ampliación de la región que engloba. Para evitar tal desviación, se puede refinar el contorno final con algunas iteraciones más y fijando el parámetro  $\alpha=0$ .



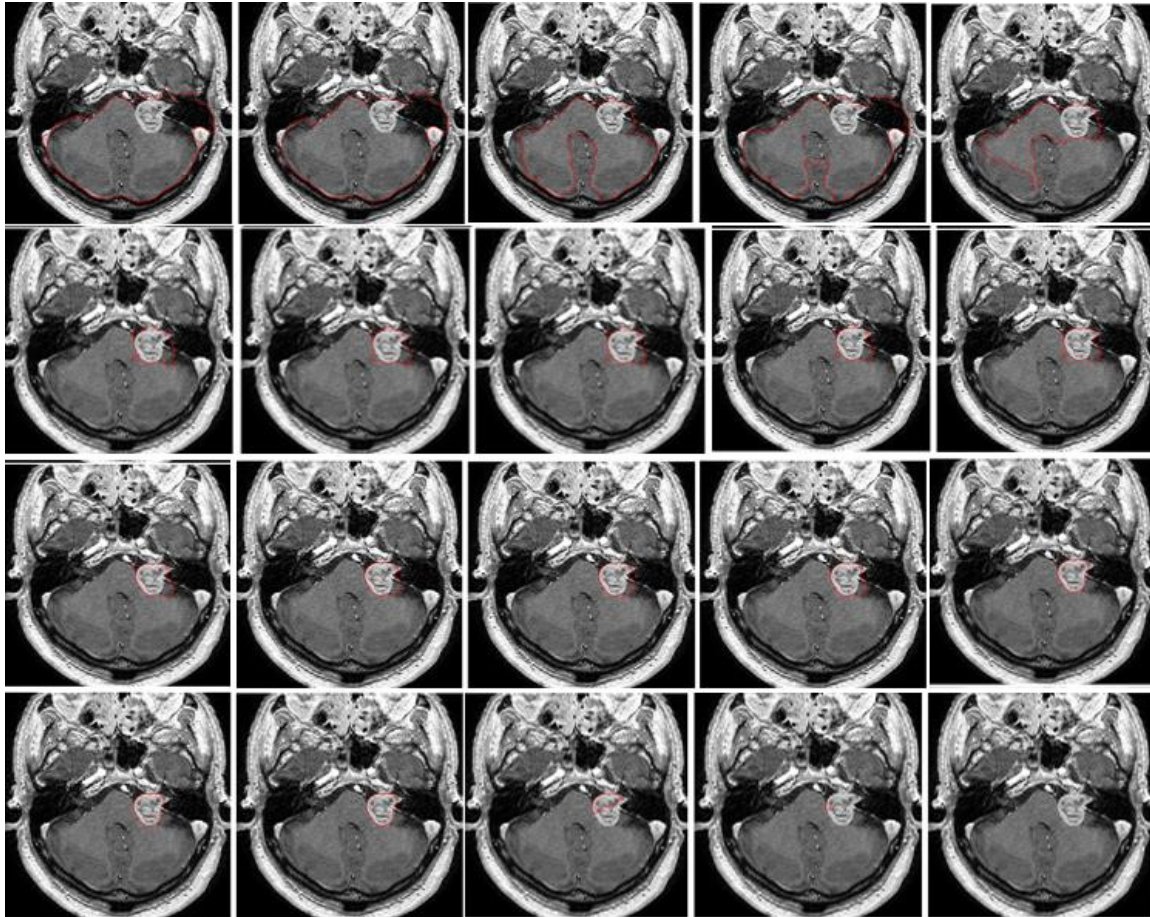
Para imágenes cuyos objetos a segmentar presenten bordes débiles, un valor grande de  $\alpha$  puede causar la curva se desborde del contorno, es decir, es más fácil que el *snake* pueda atravesar la frontera del objeto. Por lo tanto, en las imágenes con los bordes de los objetos poco definidos, el valor de  $\alpha$  debe ser pequeño.

En las Figuras 24 y 25 se muestra un ejemplo de ello. En la Figura 24 se muestra la curva inicial (*Level set* cero) que se ha marcado alrededor del tumor y en la Figura 25 el resultado de las distintas iteraciones con valores de  $\alpha$  de -10 hasta 10.



*Figura 24. Level set cero*

En la Figura 25 se aprecia como la curva va ajustándose cada vez más hasta que no detecta bien el contorno llegando a colapsar en la última iteración. Esta información sirve para saber que rango de parámetros de  $\alpha$  proporcionan un resultado válido y hacer una aproximación precisa.



*Figura 25. Progresión de la curva del snake dependiente del parámetro alpha*

Una vez se ha detectado el contorno del tumor dentro de la imagen, se quiere comprobar si se parece lo máximo posible al que ha marcado el profesional en las imágenes ROI. Para poder realizar esta comparación, se ha separado el contorno ROI del resto de la imagen, para lo cual se hace uso de los canales RGB.

Una imagen RGB tiene tres canales: rojo, verde y azul. En este contexto un canal es la imagen en escala de grises del mismo tamaño que la imagen en color, hecha de uno de estos colores primarios. Cada imagen RGB está formada a su vez por tres imágenes correspondientes a cada uno de sus canales. Como el contorno ROI está en color rojo, difiere en el valor de sus canales del resto de la imagen y esta exclusividad de tonalidad es la que se aprovecha para obtener este contorno y separarlo del fondo. En la Figura 26 se muestra un ejemplo de esto con un caso clínico diferente.

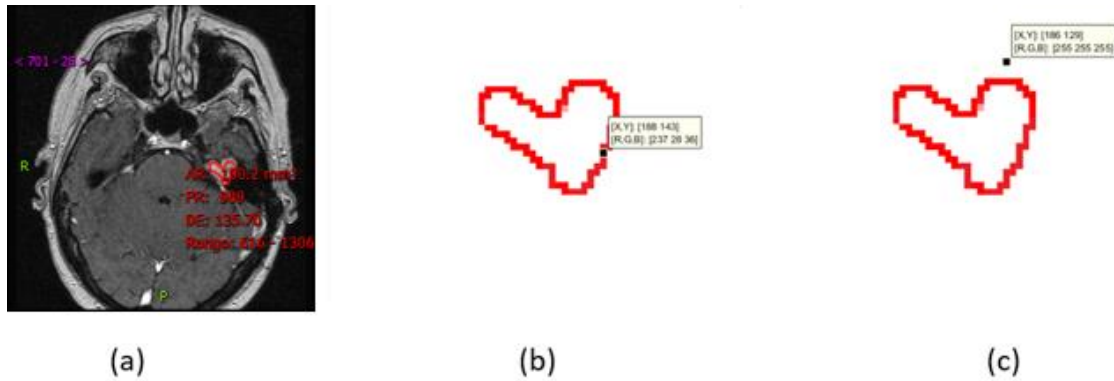


Figura 26. a) Imagen ROI original b) Canales RGB en el contorno c) Canales RGB fuera del contorno

Marcando puntos en la imagen, en el contorno y en las regiones que no son el contorno y se puede apreciar la diferencia entre canales. Los valores de RGB propios del contorno van variando, pero fuera del contorno los canales son siempre R=255, G=255, B=255 (blanco). Observando la imagen, y comparando los puntos señalados se observa como en el propio contorno el canal R puede tomar el valor de 255, pero los valores G y B no lo toman nunca. Esta diferencia se aprovecha para establecer como condición que cualquier píxel que tenga en el canal B=255 sea 0 (negro), y en caso contrario sea 1 (blanco).

De esta forma se obtiene una imagen binarizada del contorno ROI de la imagen, como se observa en la Figura 27 para hacer la transición entre la imagen b y c.

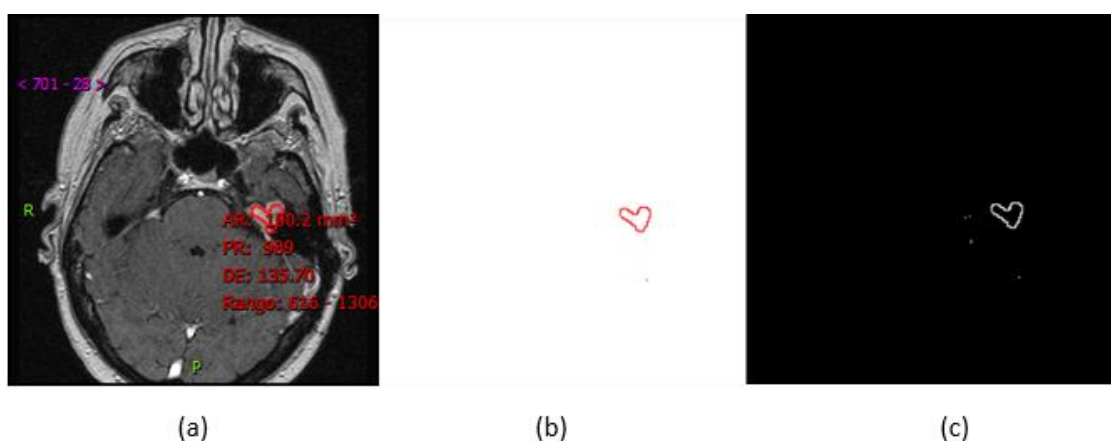


Figura 27. a) Imagen ROI original b) Contorno obtenido de la ROI original c) Contorno de la ROI original en binario





Una vez se tiene el contorno de la imagen con ROI, se toma esta imagen como referencia y se procede a compararlo con el *snake* obtenido del mismo caso clínico. Para el análisis de los resultados del *snake* obtenido respecto a los tomados como referencia, se realiza un estudio de fiabilidad a partir del análisis de la curva ROC.

### 3.2.1.1. Curva ROC

La curva ROC sirve para decidir cómo de similares son un conjunto de instancias, en este caso, imágenes. Hay cuatro posibles soluciones para el clasificador binario de la curva ROC: verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos. Estos conceptos se definen considerando el problema que se aborda en este trabajo.

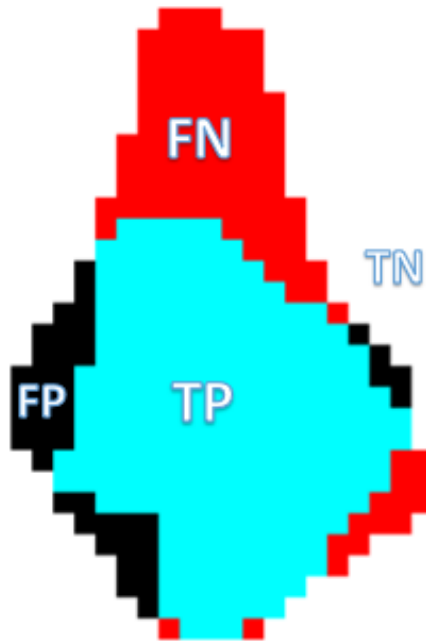
Las imágenes con las que se consta son de tipo logical, es decir, el valor de cada píxel puede ser 0 o 1 y las operaciones realizadas con las imágenes son operaciones lógicas.

- Verdaderos Positivos(TP): Aquellos píxeles que coinciden en la región del *snake* y en la región de imagen con ROI. Se puede calcular como la intersección entre las dos imágenes.
- Falsos Positivos(FP): Aquellos píxeles de la *snake* que no entran dentro de la región de la imagen con ROI. Se puede calcular como la resta lógica de la imagen de la *snake* y la obtenida de TP.
- Falsos Negativos(FN): Aquellos píxeles de la imagen con ROI que no están comprendidos en la región del *snake*. Se puede calcular como la resta lógica de la imagen con contorno ROI y la imagen TP.
- Verdaderos Negativos(TN): Aquellos píxeles que quedan fuera tanto de la región obtenida por la *snake* como de la imagen con ROI. Se puede calcular como la negación de la suma lógica de las imágenes obtenidas en TP, FP y FN.

Superponiendo el contorno de la ROI sobre el que se ha obtenido con el *snake*, se ve claramente que regiones se corresponden a los TP, TN, FP y FN (Figura 28).

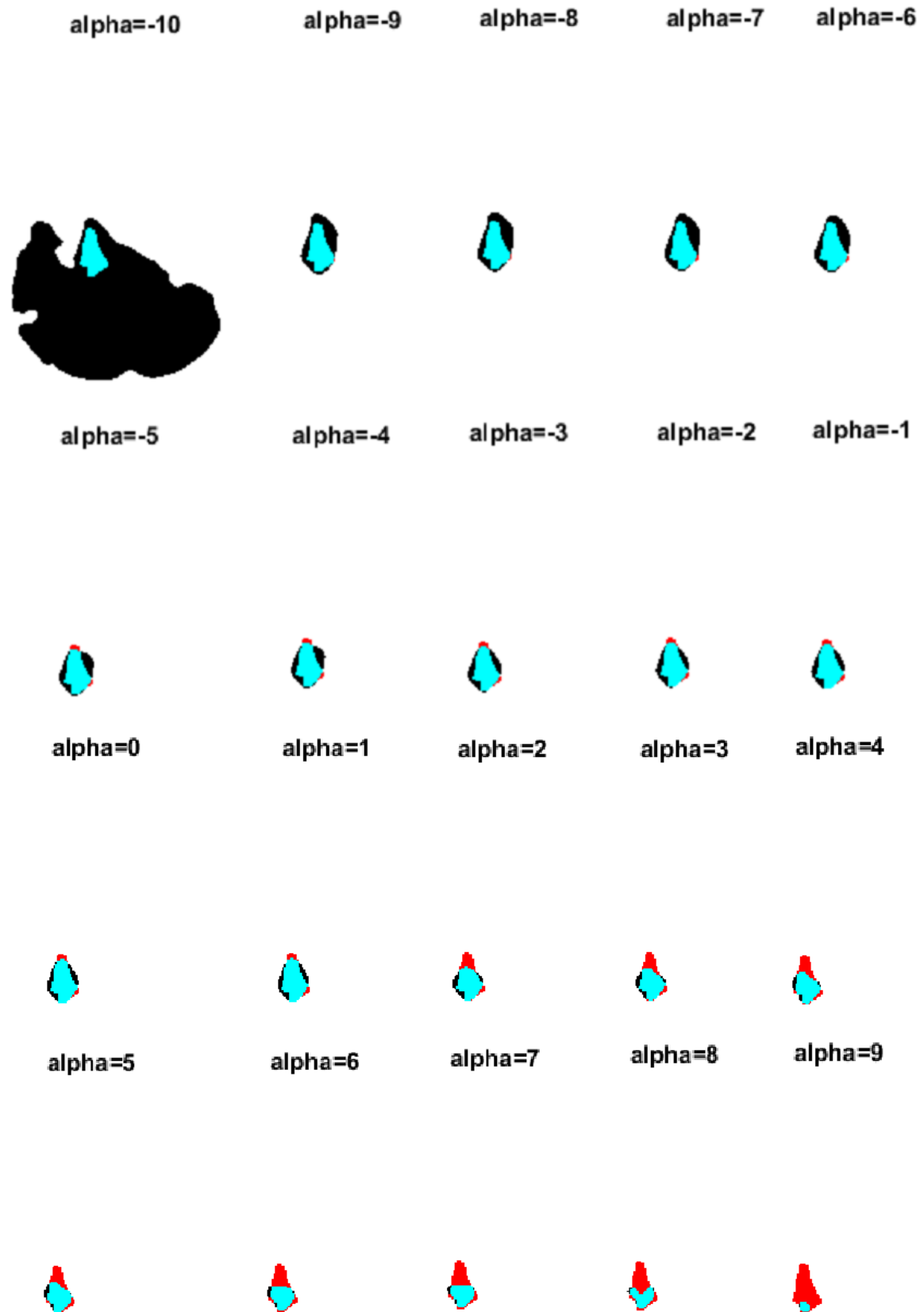


A partir de estos valores, se puede calcular la sensibilidad y fiabilidad de cada iteración del algoritmo y dibujar su curva ROC. Por ejemplo, en la Figura 28 muestra un resultado de superposición del contorno obtenido por el *snake* y la imagen ROI y se ha ampliado para poder ver las regiones mejor de forma que se puede apreciar que la región coloreada en cian se corresponde a los TP, la negra FP, la roja FN y la blanca TN.



*Figura 28. FN, TP, FP, TN de la superposición de contornos*

En la Figura 29 se muestra la evolución del contorno del ejemplo de tumor de la Figura 28, dependiendo del valor de alpha según cada iteración.



*Figura 29. Superposición de la región deseada y el Snake para cada iteración de alpha*



Con los datos obtenidos se puede calcular la sensibilidad y la especificidad de la función. Estas dos medidas se definen como:

$$\text{Sensibilidad} = \frac{TP}{TP + FN} \quad (81)$$

$$\text{Especificidad} = \frac{FP}{FP + TN} \quad (82)$$

La curva ROC se grafica como sensibilidad respecto a 1-Especificidad, La sensibilidad se corresponde con la razón de verdaderos positivos (TPR) y mide la capacidad de detectar los casos positivos correctamente. La medida 1-Especificidad, se corresponde con la razón de falsos positivos (FPR), y marca cuántos resultados positivos son incorrectos dentro de los casos negativos disponibles (Park & Goo, 2004).

Téngase en cuenta que, a partir de ahora y en las gráficas obtenidas, se llamará especificidad a 1-especificidad.

Se han recogido todos los datos necesarios en tablas con los distintos valores obtenidos de sensibilidad y especificidad en cada iteración de alpha para poder graficar la curva ROC de las imágenes disponibles, las cuales se adjuntan en el Anexo C.

A partir de los datos de sensibilidad y especificidad en cada iteración se puede representar la curva ROC, como se muestra en la Figura 30. A partir de esta gráfica se puede observar que en las primeras iteraciones con valores de alpha bajos, el valor de la sensibilidad y especificidad son altos, y conforme va aumentando el valor del parámetro alpha, la sensibilidad y especificidad va disminuyendo hasta llegar a 0.

Para evitar tener que hacer más iteraciones de las necesarias con un valor de alpha que nos proporcione una sensibilidad y especificidad de valor 0, se ha programado una condición de forma que una vez que la sensibilidad y especificidad sean 0, no se ejecuten más iteraciones.



Comparando la Figura 29 y la Figura 30, se observa como con un valor de  $\alpha=-10$  el contorno se aleja bastante de lo deseado, teniendo una gran cantidad de FP, siendo este valor de  $\alpha$  el correspondiente al punto de la gráfica con una sensibilidad de 1 y con una especificidad mayor que el resto de los puntos. Esta especificidad elevada es debido a la cantidad de FP.

En el rango de  $\alpha=-9$  hasta  $\alpha=-6$ , el contorno del *snake* apenas varía, y se aproxima más al contorno que en las iteraciones anteriores, disminuyendo considerablemente el número de FP. Estos cuatro puntos se representan en la curva ROC con un valor alto de sensibilidad y un valor muy bajo de especificidad.

En el rango de valores de  $\alpha=-5$  hasta  $\alpha=1$ , ha disminuido el número de píxeles FP, y empiezan a aparecer FN. Igualmente, la evolución de la curva es mínima, lo cual ofrece un rango bastante grande de valores de  $\alpha$  con resultados aproximados al contorno que se desea segmentar, ofreciendo robustez al algoritmo. En la curva ROC este rango de valores se representa con una sensibilidad media y un valor muy bajo de especificidad.

Por último, en el rango de  $\alpha=2$  hasta  $\alpha=9$ , el número de píxeles con FN va aumentando, hasta que acaba cubriendo toda la zona, esto se debe a que el *snake* ha colapsado. En la curva ROC se representa con valores de sensibilidad medios y bajos y especificidad muy bajos.

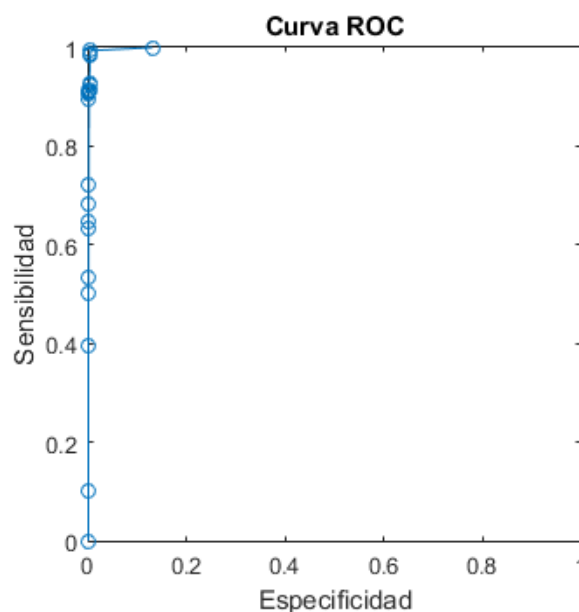


Figura 30. Curva ROC



A partir de la curva obtenida se puede calcular el área bajo la curva (AUC). Este valor sirve para medir la precisión de la prueba, de forma que esta precisión se puede interpretar a partir de unos intervalos para los resultados de los valores de AUC.

[0.5, 0.6): Mala precisión.

[0.6, 0.75): Precisión regular.

[0.75, 0.9): Buena precisión.

[0.9, 0.97): Precisión muy buena

[0.97, 1): Precisión excelente.

Para calcular el área bajo la curva se utiliza el muestreo de la gráfica con cada iteración de alpha, en las que se ha dibujado un punto(x,y) donde x es el valor de especificidad e y el valor de la sensibilidad. A partir de estos datos, se calcula el área comprendida entre un punto y el siguiente más la mitad del intervalo siguiente, de forma que, sumando cada pequeña área se obtiene el área final bajo la curva. Esta explicación puede entenderse mejor observando la Figura 31 y la Tabla 2.

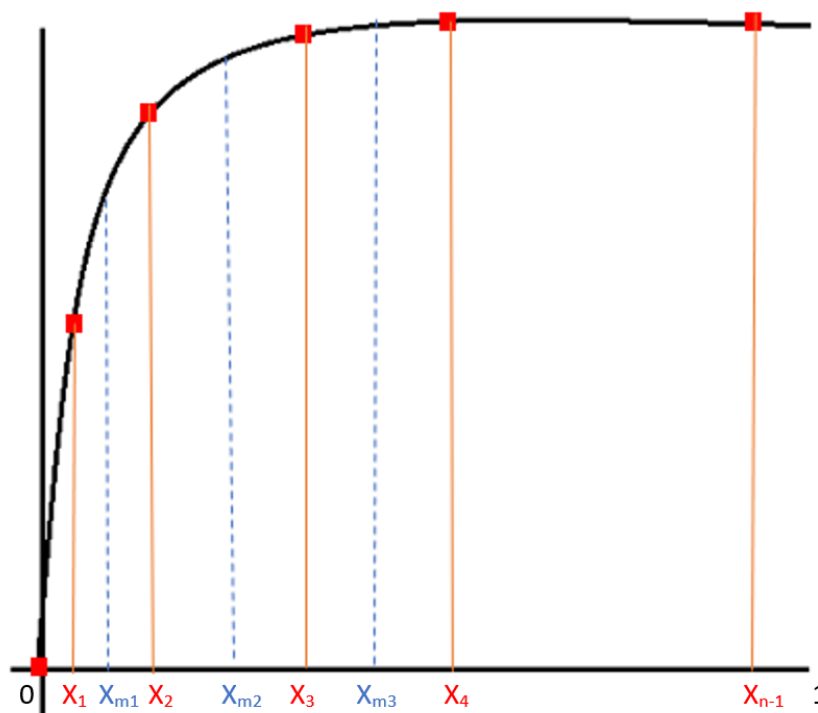


Figura 31. Método para calcular el AUC



Tabla 2. Cálculo de AUC

$X_i$	$X_{mi}$	Intervalo	$\Delta X_i$
$X_1$	-	$[0, X_{m1}]$	$X_{m1}$
$X_2$	$\frac{X_2 - X_1}{2}$	$[X_{m1}, X_{m2}]$	$[X_{m2} - X_{m1}]$
$X_3$	$\frac{X_3 - X_2}{2}$	$[X_{m2}, X_{m3}]$	$[X_{m3} - X_{m2}]$
...	...	...	...
$X_n$	$\frac{X_n - X_{n-1}}{2}$	$[X_{mn-1}, 0]$	$[1 - X_{mn-1}]$

De forma que el área bajo la curva se calcula,

$$AUC = \sum \Delta X_i Y_i \quad (83)$$

Donde  $X_i$  se corresponden con los valores de especificidad y  $Y_i$  con los valores de la sensibilidad.

El desarrollo sistemático del parámetro alpha obtenido en el caso clínico anterior y representado en la Figura 29, es un caso bastante favorable, ya que el *snake* se ha ajustado rápidamente al contorno objetivo y en su representación de la curva ROC (Figura 30) se ha obtenido un área bajo la curva de valor 0.9995, es decir, una precisión excelente.

Para comparar otros resultados, se estudia un caso menos favorable donde se ha obtenido una curva ROC representada en la Figura 32, obteniéndose un AUC= 0.9960, aún con precisión excelente, pero algo peor comparado con algunos de los otros casos.

En el desarrollo sistemático del parámetro alpha representado en la Figura 33, se puede observar como en el rango de valores de alpha=-10 hasta -6 el número de FP es muy alto, lo cual se representa en la curva ROC con unos valores muy altos de sensibilidad y altos en especificidad.

En el rango de alpha= -5 hasta alpha=0, ha disminuido el número de FP respecto a las iteraciones anteriores y además conforme van aumentando las iteraciones



la cantidad de FP disminuye. En la curva ROC del caso, hay un punto en el que el valor de la sensibilidad ha disminuido bruscamente y la especificidad no tanto, separándose de los ejes respecto al resto de los puntos, el motivo se debe a que en la iteración de  $\alpha=-5$  repentinamente la cantidad de FP disminuye, predominando los TP y ajustándose bastante mejor al contorno.

En el rango de  $\alpha=1$  hasta  $\alpha=5$ , el número de FN ha aumentado considerablemente hasta acabar ocupando el contorno totalmente, es decir, el *snake* ha colapsado. En la curva ROC se representa con los puntos con sensibilidad media y baja y especificidad muy baja.

Comparando este caso con el anterior (Figura 30 y Figura 32), se puede observar que la curva ROC del caso más desfavorable (Figura 32) tiene más valores de sensibilidad y especificidad altos debido a que hay más iteraciones de  $\alpha$  con alto nivel de FP. En ambos se aprecia que conforme va aumentando el valor de  $\alpha$ , menor es la sensibilidad y especificidad, hasta que llega a un valor nulo de ambas, lo cual ocurre cuando toda la región deseada está comprendida por FN. Sin embargo, la conclusión más importante al comparar estos dos casos es que en el caso más favorable el rango de valores de  $\alpha$  que proporcionaban un resultado aproximado al contorno deseado estaba comprendido entre -9 y 2, y en el caso menos favorable de -1 hasta 5. Es decir, el rango de valores admisibles de  $\alpha$  ha disminuido considerablemente, y por ello es importante elegir un parámetro valor óptimo de  $\alpha$ .

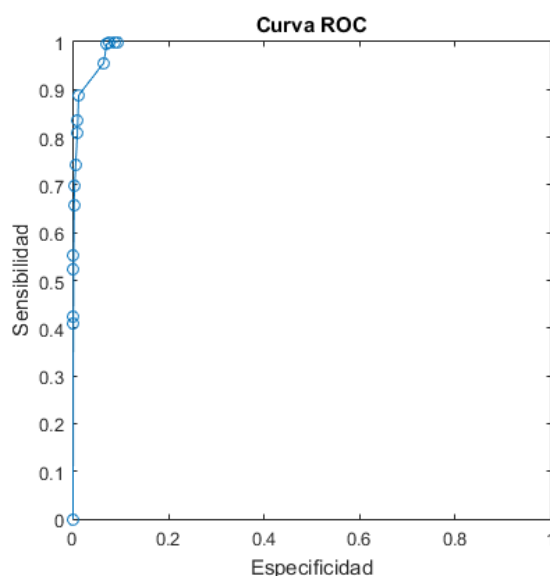


Figura 32. Curva ROC segundo caso



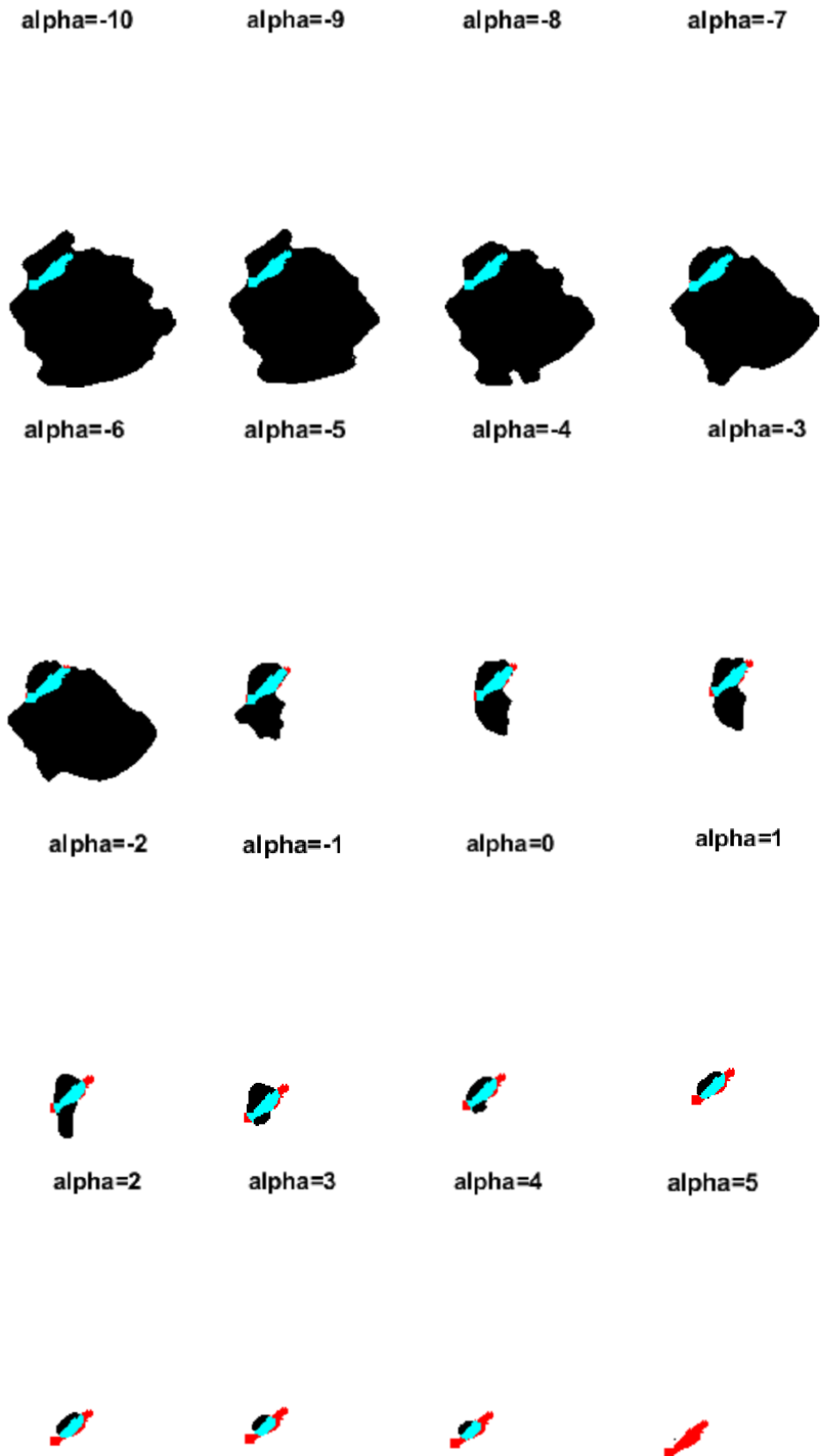


Figura 33. Comparación de contornos segundo caso



### 3.2.2. Cálculo del valor óptimo de alpha

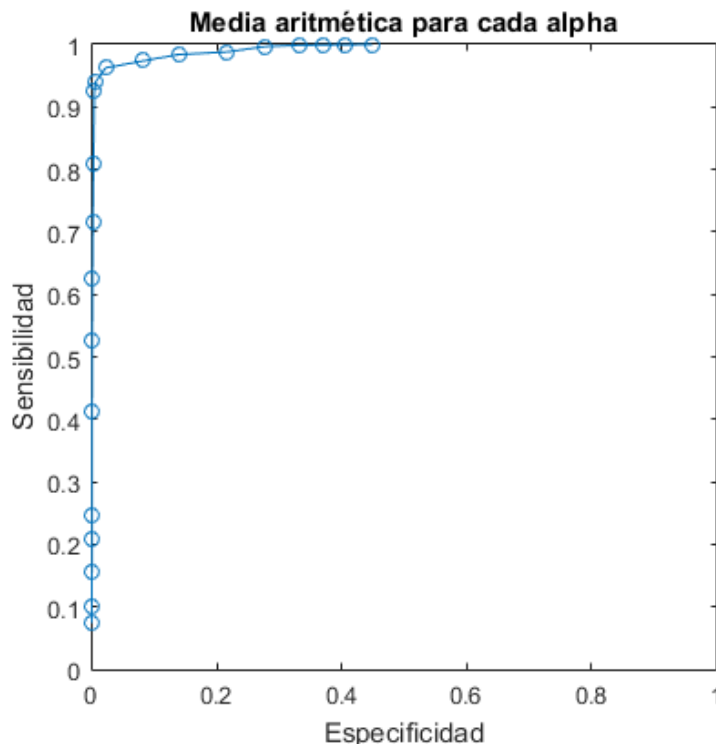
La obtención del valor óptimo de alpha es importante para obtener unos buenos resultados de segmentación sin la necesidad de variar este parámetro en todos los casos clínicos con los que se cuenta. Para ello, se realiza una media ponderada de la sensibilidad y la especificidad obtenida para cada iteración de alpha, dadas las tablas que se encuentran en el Anexo E, de forma que se obtienen los resultados de la tabla 3,

*Tabla 3. Valores medios de sensibilidad y especificidad por cada valor de alpha*

alpha	sens	espec2
-10	1	0,4480
-9	0,9996	0,4050
-8	0,9992	0,3696
-7	0,9988	0,3323
-6	0,9966	0,2769
-5	0,9879	0,2165
-4	0,9838	0,1386
-3	0,9737	0,0830
-2	0,9624	0,0219
-1	0,9394	0,0058
0	0,9250	0,0041
1	0,8083	0,0026
2	0,7164	0,0017
3	0,6256	0,0011
4	0,5274	0,0008
5	0,4138	0,0004
6	0,2457	0,0002
7	0,2083	0,0002
8	0,1548	0,0001
9	0,1016	0,0001
10	0,0739	0,0001



A partir de estos datos se puede representar la curva ROC (Figura 34),



*Figura 34. Curva ROC para valores medios*

Con esta representación de la curva ROC se puede calcular el área bajo la curva (AUC), de forma que se ha obtenido  $AUC = 0.9956$ , es decir, un valor excelente en el rango de precisión de la curva ROC.

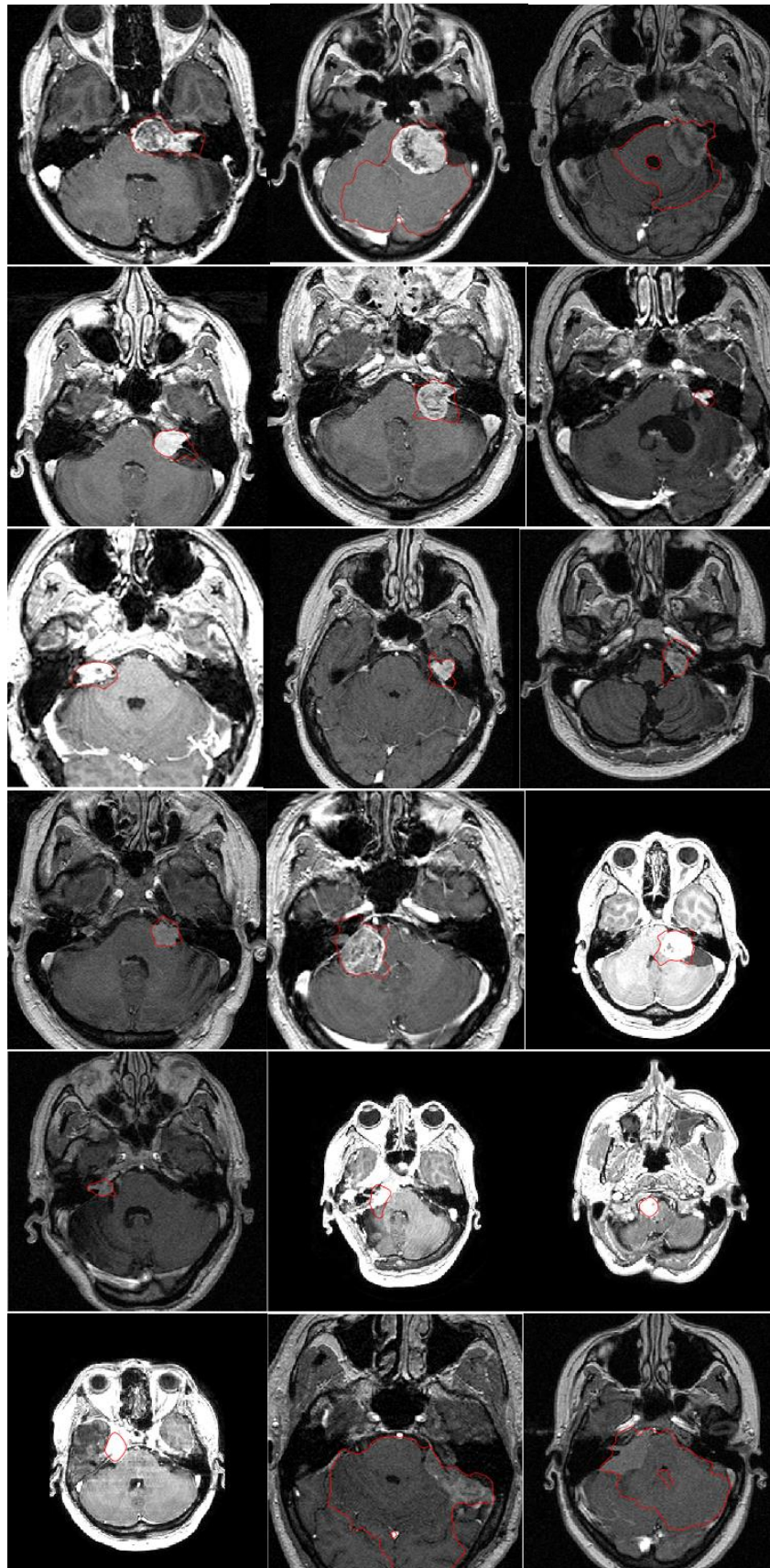
Para escoger el valor óptimo de alpha, se escoge una iteración cuyo valor de sensibilidad sea alto y su especificidad bajo, de forma que se obtiene una tasa de error baja. Si se observa la tabla 3, se puede elegir  $\alpha = -1$  como el valor óptimo de alpha, con una sensibilidad de 0.9394 y especificidad 0.0058, debido a que parece que la tasa de error de sensibilidad respecto a la especificidad es la menor posible.

Para poder analizar mejor la exactitud del método, primero se va a analizar los resultados con una iteración cercana al parámetro que se ha decidido como óptimo, por ejemplo,  $\alpha = -2$  con un valor de sensibilidad 0.9624 y de especificidad 0.0219, que respecto con el valor  $\alpha = -1$  tomado como óptimo aumenta ligeramente la sensibilidad pero a la vez su especificidad es mayor, lo cual provoca una mayor tasa de error.

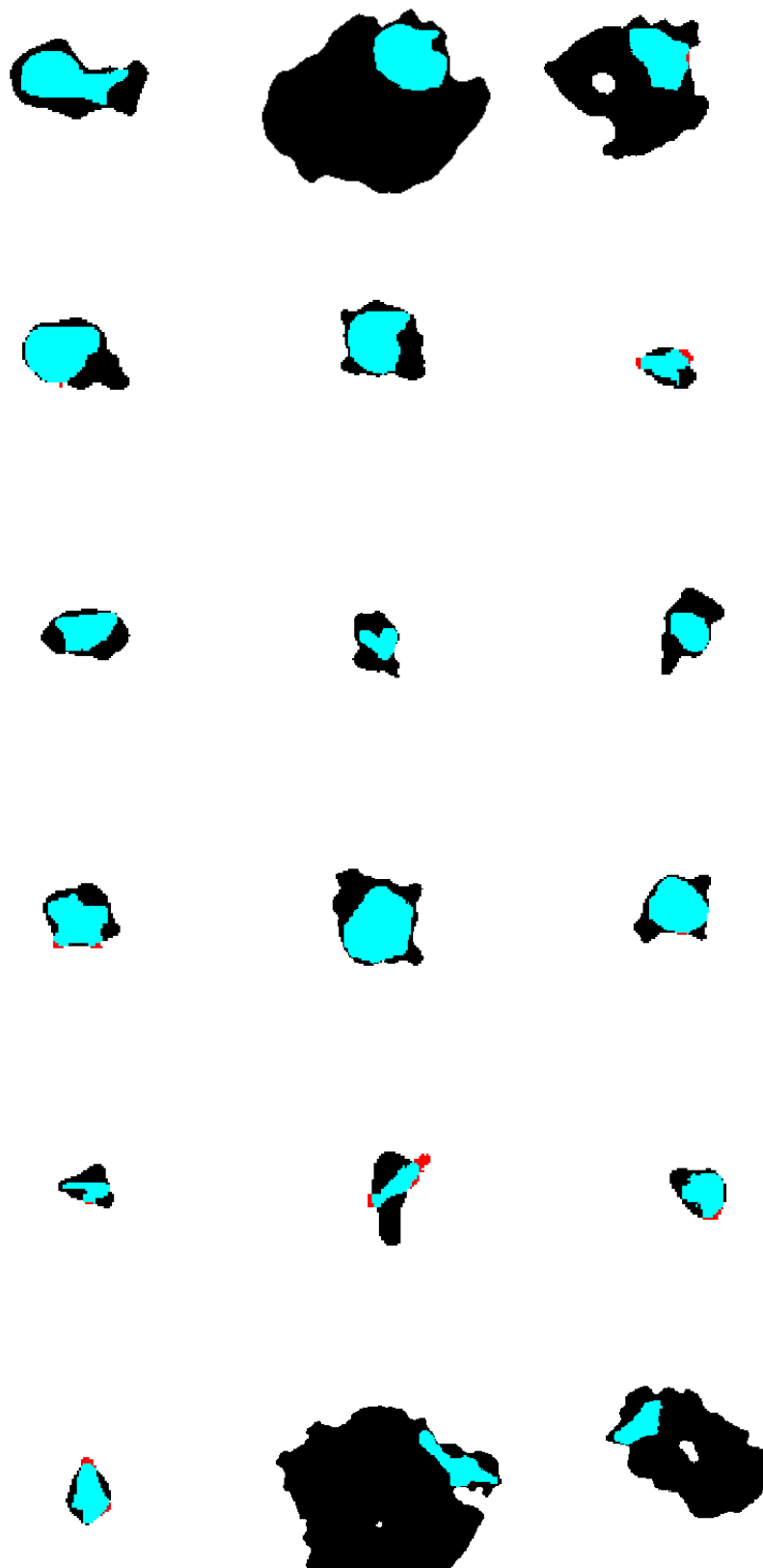


Se elige el valor  $\alpha = -2$  por ser un valor cercano al elegido como óptimo, y proporcionar además unos buenos valores de sensibilidad y especificidad, de manera que la comparación de los resultados obtenidos con ambos valores proporcione resultados más robustos que justifiquen la elección del parámetro óptimo  $\alpha = -1$ .

En la Figura 35, se representa la curva del Snake para cada caso con el parámetro de  $\alpha = -2$ , la cual se compara con el contorno deseado en la Figura 36. Se observa que en algunos casos los resultados son bastante buenos, se ajustan bastante al contorno objetivo, pero en otros casos la cantidad de falsos positivos es bastante grande.



*Figura 35. Snake obtenido con  $\alpha=-2$  en cada caso*



*Figura 36. Superposición de la snake con  $\alpha=-2$  y el contorno deseado*



En un caso favorable, comparando la iteración de  $\alpha = -2$  con el valor elegido como óptimo, es decir,  $\alpha = -1$ , se observa que el contorno del *snake* apenas varía (Figura 37), debido la curva ha evolucionado y ya se adapta correctamente al contorno deseado, y tal y como se ha deducido teóricamente, al llegar a un contorno parecido al objetivo, el *snake* se ralentiza.

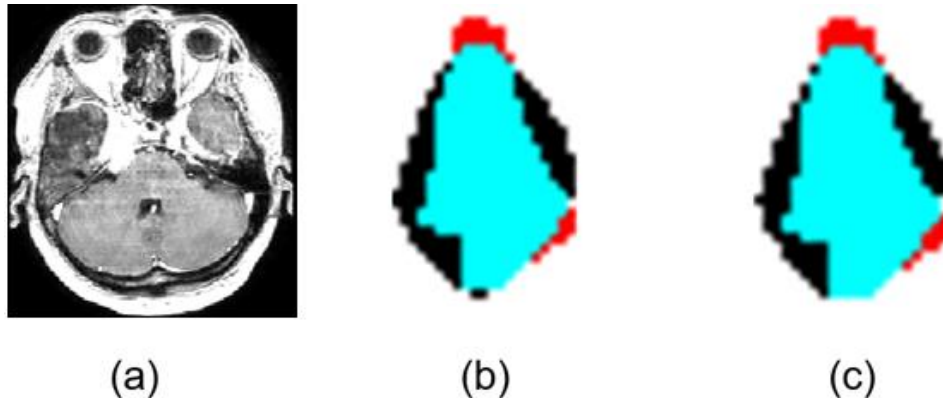


Figura 37. Superposición de contornos a) imagen original b)  $\alpha = -2$   
c)  $\alpha = -1$

En un caso desfavorable, comparando la iteración de  $\alpha = -2$  con el valor elegido como óptimo, es decir,  $\alpha = -1$ , se observa que el contorno del *snake* varía considerablemente (Figura 38), esto es debido la contraposición de lo anterior, es decir, como el *snake* está lejos del contorno deseado, la curva evoluciona rápidamente.

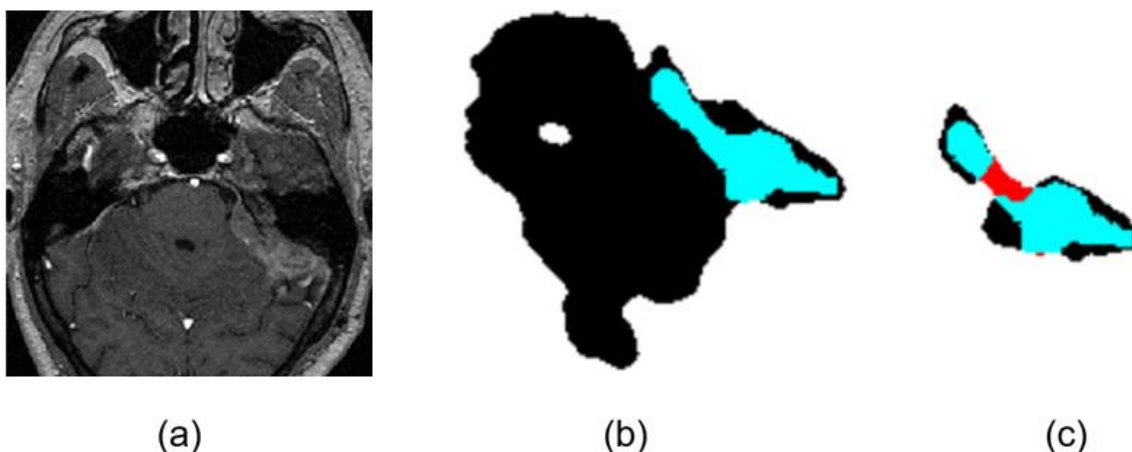
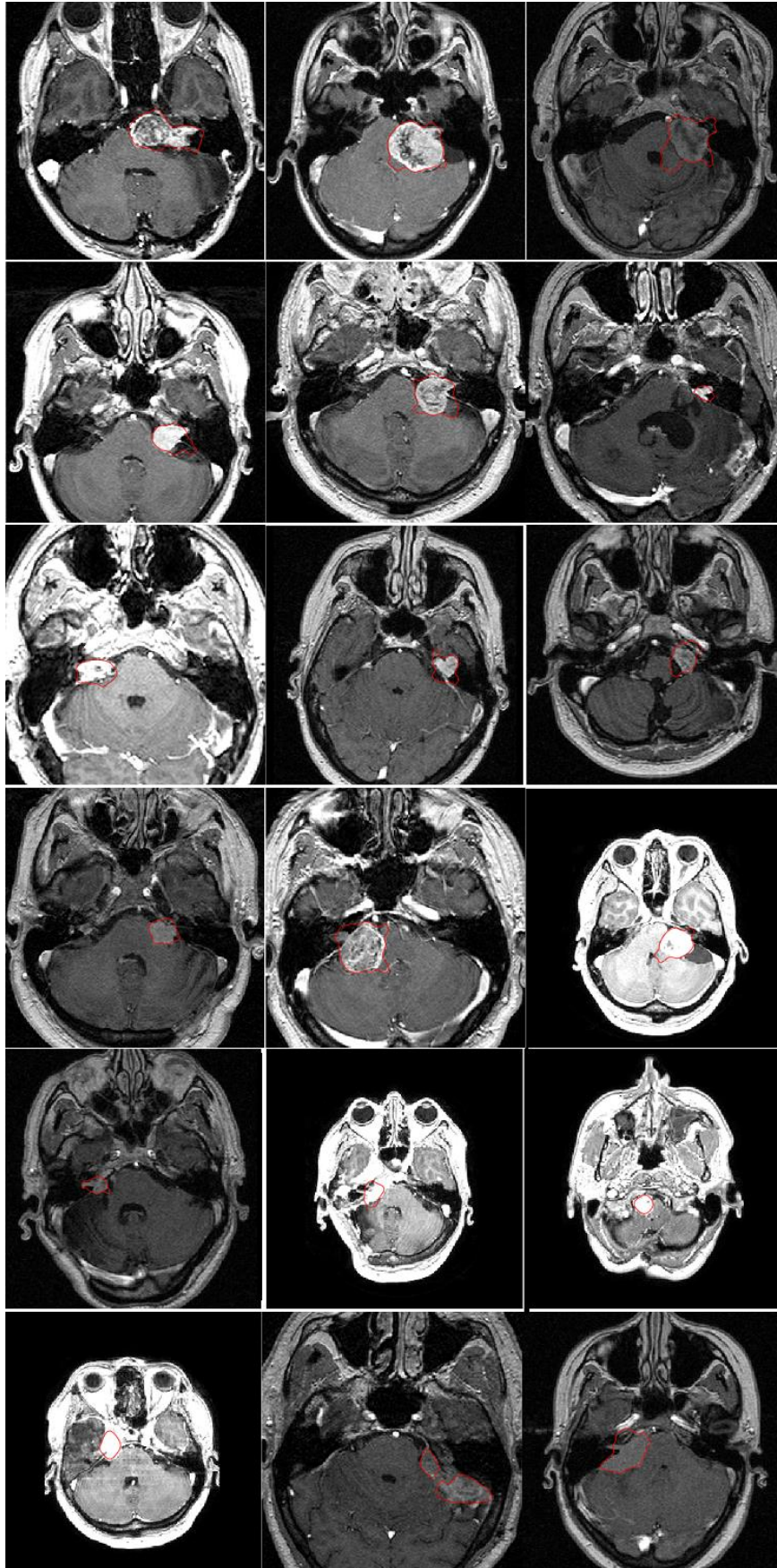


Figura 38. Superposición de contornos a) imagen original b)  $\alpha = -2$   
c)  $\alpha = -1$

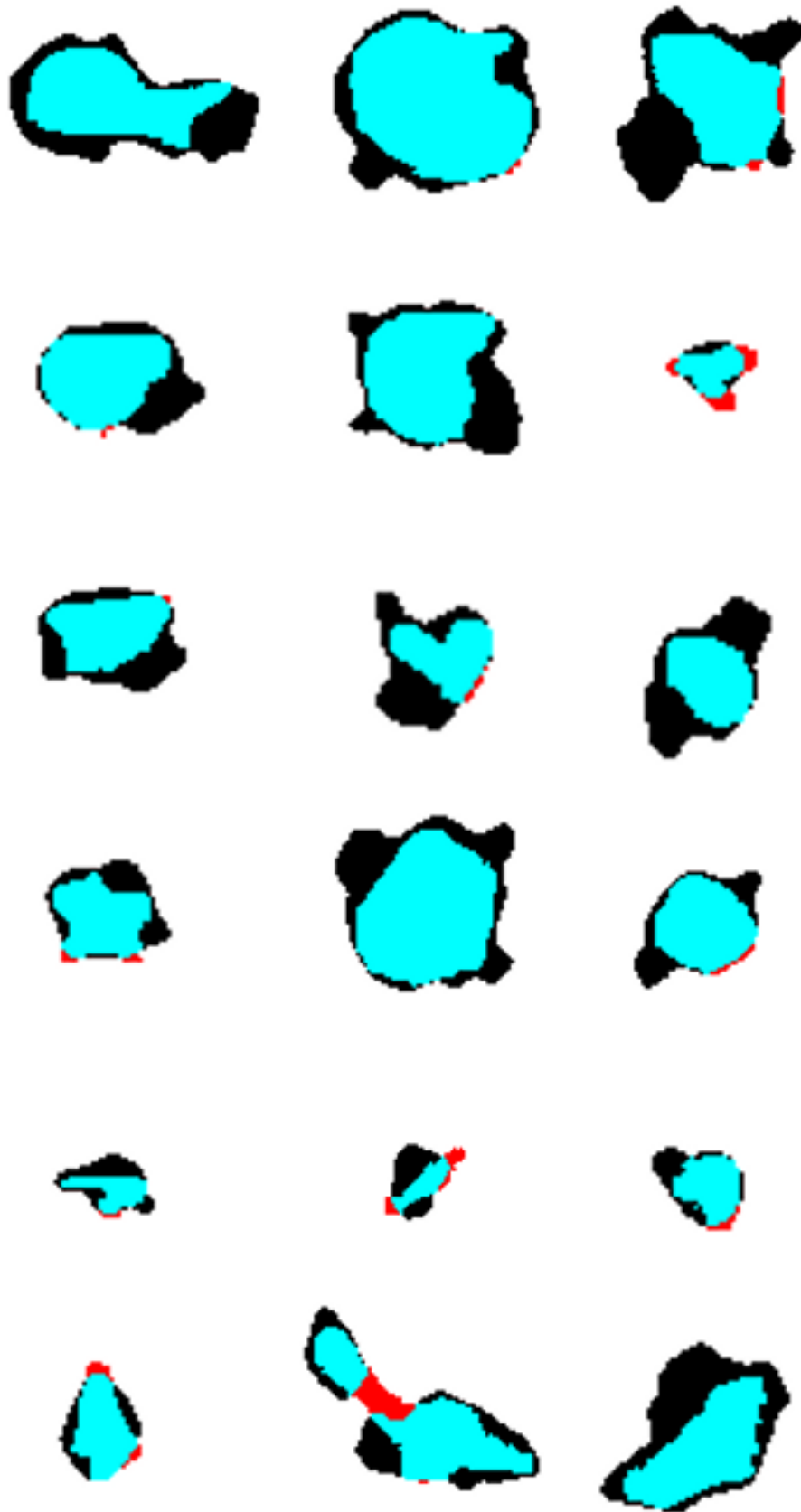


Los resultados que se han obtenido con el valor óptimo de  $\alpha=-1$  se representan en la Figura 39 y 40.





*Figura 39. Snake obtenido con  $\alpha=-1$  para cada caso*



*Figura 40. Superposición de la snake con  $\alpha=-1$  y el contorno deseado*



Los resultados obtenidos con el valor de alpha estimado óptimo se ajustan bastante bien al contorno aunque en la mayoría de ellos hay una cantidad de FP, y algunos casos presentan zonas con FN. Sin embargo, comparando con el caso anterior en la iteración  $\alpha=-2$ , los resultados han mejorado bastante con sólo una iteración más.

### 3.3. RESULTADOS OBTENIDOS

La conclusión de estos resultados es que esta evolución se debe a, como se dice en la base teórica de DRSLE (Li, Xu, Gui, & Fox, 2010), la función  $A_g$  que acompaña a  $\alpha$  acelera el movimiento del *Level Set* cero en el proceso de evolución del nivel establecido. Cuando el contorno inicial se sitúa lejos de los bordes deseados, acelera la ampliación o la reducción del contorno nivel cero. Cuando está cerca del contorno lo ralentiza tomando  $g$  valores más pequeños. Esto es debido a que en localizaciones lejanas al nivel cero el gradiente es cero y en las proximidades 1.

Comparando los resultados obtenidos del contorno activo desarrollado aplicando DRSLE y el contorno de referencia, los resultados de la medición del área bajo la curva (AUC) de la curva ROC son de una precisión muy buena o excelente, lo que demuestra que el algoritmo funciona de la manera esperada y se ajusta al contorno deseado. La representación de la curva ROC de todos los casos clínicos se incluyen en el Anexo C, así como los resultados obtenidos de AUC se encuentran en el Anexo D.

El estudio de la influencia del parámetro  $\alpha$  en el *snake* es necesario para encontrar un valor óptimo de este que pueda utilizarse para el resto de casos clínicos y obtener buenos resultados. Se han logrado resultados aceptables, con una cantidad de FP y FN baja respecto a la cantidad de TP en la superposición de contornos, que podrían mejorarse si se haya el valor óptimo del resto de parámetros que condicionan la curva. La fiabilidad del parámetro  $\alpha$  óptimo escogido se ha demostrado con la medición de AUC de la curva ROC de la media aritmética de todos los casos. Además, se ha demostrado que el valor de  $\alpha$  elegido como óptimo mejora la tasa de error respecto a un valor de  $\alpha$  cercano, que daba resultados de sensibilidad y especificidad bastante buenos.



### 3.4. CONCLUSIONES

Tras la realización de este trabajo la principal conclusión que se obtiene es que fue posible desarrollar una metodología para segmentar estructuras cerebrales mediante el método de contornos activos, donde se han obtenido resultados de calidad, con una fiabilidad según el test de precisión de la curva ROC excelente. Esto significa que se han cumplido los objetivos de este trabajo.

Se ha realizado un estudio de distintos métodos de contornos activos, y finalmente el método de formulación que se ha utilizado ha sido los contornos activos basados en *Level Set*, en concreto en método basado en DRSLE debido a que esta formulación mantiene la regularidad de la LSF, con un cálculo preciso y una evolución estable del *Level Set*. La formulación DRLSE permite el uso de pasos de tiempo relativamente grandes para reducir significativamente los números de iteración y el tiempo de cálculo, manteniendo suficiente precisión numérica en ambos casos debido a la regularización.

Además, cabe destacar que hay varios factores que influyen sobre el *snake*, siendo el más influyente el parámetro  $\alpha$ . Se ha realizado un desarrollo sistemático de este parámetro en el que se muestra como varía el *snake* según el valor que tome para encontrar su valor óptimo que aproxime el contorno en todos los casos. Encontrar un valor óptimo del parámetro  $\alpha$  es importante porque con valores demasiado bajos, el *Snake* presenta una gran cantidad de FP respecto al contorno deseado, es decir, el *snake* está lejos del contorno objetivo. Por otro lado, si el valor de  $\alpha$  es demasiado alto, la cantidad de FN aumenta hasta que la curva del *snake* se desestabiliza. Por ello, se ha realizado una media ponderada de todos los valores de sensibilidad y especificidad de todos los casos en cada iteración de  $\alpha$  y se ha estimado un valor óptimo que proporciona una menor tasa de error.

El estudio de la influencia del parámetro  $\alpha$  en el *snake* es necesario para encontrar un valor óptimo de este parámetro de forma que pueda usarse, al igual que se ha fijado el resto de parámetros dependientes de la curva, como un valor constante que aproxime la curva del *snake* al contorno deseado.



Otro factor influyente en los resultados del *snake* es la definición del contorno inicial, ya que, si está muy alejado de la zona a segmentar, la curva no va a evolucionar bien debido a que se puede ver influenciada por otras estructuras. Cuanto más parecido sea el contorno inicial a la zona a segmentar, el método proporcionará mejores resultados.

Por lo tanto, es importante determinar correctamente los parámetros que influyen en la evolución de la curva, ya que cada uno provoca unas consecuencias en la ecuación del *snake*, y debido a que las imágenes con las que se cuenta son en escala de grises, algunas de ellas con diferencias muy pequeñas de intensidad, los valores de los parámetros deben ser precisos para obtener unos resultados adecuados.

Se destaca también la versatilidad y facilidades que ofrece el software MATLAB para el procesamiento de imágenes, permitiendo el tratamiento de imágenes médicas en formato DICOM de forma sencilla, además de su lenguaje M, que permite desarrollar un código intuitivo de métodos matemáticos complejos como los que se abordan en este trabajo, además de las operaciones y funciones que se pueden realizar con las imágenes.

### **3.5. FUTURAS LÍNEAS DE INVESTIGACIÓN**

Como líneas futuras sería interesante profundizar sobre otros tipos de contornos activos o mejorar la implementación del método utilizado. Este método puede mejorarse haciendo un estudio sistemático de no sólo el parámetro  $\alpha$ , si no también de todos los parámetros que influyen en los resultados del *snake*. Con estos, si cabe, se podría mejorar los resultados.

Además, se podrían añadir al método nuevos parámetros que ajusten más la curva al contorno deseado, como introducir parámetros de textura. Se podría realizar un estudio de texturas de la imagen, de forma que, debido a la diferencia de texturas entre la zona interna y la zona externa del objeto a segmentar, las energías externas e internas que afectan a la evolución de la curva del *snake* se verían influenciadas también por estos nuevos parámetros.





Otra mejora que se propone es reducir el tiempo de ejecución del programa, mejorando la eficiencia del algoritmo y sobre todo de la búsqueda del parámetro óptimo de alfa, ya que, al necesitar muchas iteraciones, desemboca en un costo de tiempo de ejecución.

En este trabajo se han obtenido unos buenos resultados y además se ha obtenido un valor óptimo del parámetro alpha con los casos clínicos dados, pero habría que ver cómo funciona en el caso de tener múltiples y distintos casos clínicos distintos.

En este caso concreto, se podría también ampliar el estudio a partir de estos resultados, de forma que, si se tuviera las imágenes de resonancia magnética de la evolución de un paciente, se podría ver como evoluciona el tumor, y haciendo comparaciones entre una imagen y otra se podría calcular la velocidad con la cual crece el tumor. Además, se podrían hacer estudios en 3D, calcular el volumen del tumor, su malignidad o benignidad según la heterogeneidad de sus bordes, etcétera.

Para finalizar, cabe destacar que el método utilizado puede tener diversas aplicaciones además de las que se han desarrollado en este trabajo. La finalidad del método de contornos activos es encontrar una zona que se ha definido previamente, por lo que se puede utilizar para otras aplicaciones, dentro de la medicina algunos ejemplos podrían ser la segmentación de otras zonas del cerebro y distintas zonas del cuerpo además de la segmentación de arterias y venas en el sistema circulatorio. No sólo se puede realizar en imágenes por resonancia magnética, si no de cualquier tipo. Otras aplicaciones pueden ser la identificación y extracción de objetos o seres vivos en diferentes áreas para tratar esa zona de la manera que se desee.



# Bibliografía

- ABTA, A. B. (2012). *ABTA*. Obtenido de [www.abta.org](http://www.abta.org)
- Bar, L., Chan, T. F., Chung, G., Jung, M., Kiryati, N., Mihieddine, R., . . . Vese, L. A. (2011). Mumford and Shah Model and its applications to Image Segmentation and Image Restoration. En *Handbook of mathematical methods in imaging* (págs. 1095-1157). Springer New York.
- Caselles, V., Kimmel, R., & Sapiro, G. (1997). Geodesic active contours. *International journal of computer vision*, 22(1), 61-79.
- Chan, T., & Vese, L. (1999). An active contour model without edges. *Int. Conf. ScaleSpace Theories in Computer Vision*, 141-151.
- Chan, T., & Vese, L. (2001). Active contours without edge. *IEEE Transactions on image processing*, 10(2), 266-277.
- Coto, E. (2003). Métodos de segmentación de imágenes médicas. *Universidad Central de Venezuela: Lecturas en Ciencias de la Computación*, 9-15.
- Forero, M., Sarmiento, W., & Díaz, G. (2001). Aplicación para la lectura y conversión de formatos DICOM no estándar y génesis. *II Congreso Latinoamericano de Ingeniería Biomédica*.
- Gonzalez, R. (2016). *Digital Image Processing*.
- Hernández Olvera, E. C., & Narario Alvarez, J. C. (2011). Eliminación de ruido y segmentación de formas en imágenes médicas. Universidad Autónoma de México. Facultad de ingeniería, México. Obtenido de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/171/A5.pdf?sequence=5>
- Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision*, 321-331.



- Lamas Rodríguez, J., Quesada Barriuso, P., Argüello, F., Heras, D. B., & Bóo, M. (2012). Proyección del método de segmentación del conjunto de nivel en GPU.
- Li, C., Xu, C., Gui, C., & Fox, M. D. (2010). Distance Regularized Level Set Evolution and Its. *IEEE transactions on image processing*, 19(12), 3243-3254.
- MathWorks. (1994-2018). Obtenido de <https://es.mathworks.com/products/matlab.html>
- Mildenger, P., Eichelberg, M., & Martin, E. (2002). *E. Eur Radiol*. Obtenido de <https://doi.org/10.1007/s003300101100>
- Mitiche, A., & Ben Ayed, I. (2011). *Variational and Level Set Methods in Image Segmentation*. Springer.
- Mumford, D., & Shah, J. (1985). Boundary detection by minimizing functionals. *Proceedings of IEEE conference on computer vision and pattern recognition*, 22-26.
- Mumford, D., & Shah, J. (1989). *Optimal approximations by piecewise smooth functions and associated variational problems*.
- Nicoguardo. (2018). *Wikimedia Commons*. Obtenido de [https://commons.wikimedia.org/wiki/File:Level\\_set\\_method.png#/media/File:Level\\_set\\_method.png](https://commons.wikimedia.org/wiki/File:Level_set_method.png#/media/File:Level_set_method.png)
- Olabe, X. B. (1998). *Redes Neuronales Artificiales y sus Aplicaciones. Publicaciones de la Escuela de Ingenieros 101pp*.
- Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 12-49.
- Palomino, N. L., & Concha, U. R. (2009). Técnicas de segmentación en procesamiento digital de imágenes. *Revista de investigación de Sistemas e Informática*, 6(2),9-16.





- Park, S., & Goo, J. J. (2004). *Receiver Operating Characteristic (ROC) Curve: Practical Review for Radiologists*. Obtenido de <https://doi.org/10.3348/kjr.2004.5.1.11>
- Pérez Segura, P. (2017). *SEOM*. Obtenido de <https://www.seom.org/es/info-sobre-el-cancer/tumores-cerebrales?format=pdf>
- Real Academia Española[RAE]. (2017). *Diccionario de la lengua española* (23.<sup>a</sup> ed.). Consultado en <http://www.rae.es/rae.html>
- Sinning, M. (2017). Clasificación de los tumores cerebrales. *Revista Médica Clínica Las Condes*. Obtenido de <http://www.elsevier.es/es-revista-revista-medica-clinica-las-condes-202-articulo-clasificacin-de-los-tumores-cerebrales-S0716864017300597>
- Vélez, J. F., Sánchez, Á., Moreno, A. B., & Esteban, J. L. (2005). Verificación Off-Line de Firmas Manuscritas: Una Propuesta basada en Snakes Paramétricos. In *Proc. Spanish Symp. on Shape Recognition and Artificial Intelligence (SNRFAI)*.
- Vese, L. A., & Guyader, C. L. (2015). *Variational methods in image processing*. CRC Press.
- Wainschenker, R., & J. Massa, P. T. (2011). *Etapas del procesamiento digital de imágenes. Guía informativa área Procesamiento de Señales*.
- Yu, Y., & Huang, W. (2017). Selection of the Regularization Parameter in the Ambrosio-Tortorelli Approximation of the Mumford-Shah Functional for Image Segmentation. *arXiv preprint. arXiv:1706.06459*.





# Anexos

## ANEXO A. CÓDIGO EN MATLAB

### Función DRSLE

```
function phi = drlse_edge(phi_0, g, lambda,mu, alfa, epsilon, timestep, iter,
potentialFunction)
% Este código MATLAB implementa un modelo de contornos activos como
una
% aplicación de DRSLE en el artículo de los autores C. Li, C. Xu, C. Gui, M.
D. Fox, "Distance Regularized Level/ Set Evolution and Its Application to
Image Segmentation",
% IEEE Trans. Image Processing, vol. 19 (12), pp.3243-3254, 2010.
%
% Author: Chunming Li, all rights reserved
% E-mail: lchunming@gmail.com
% li_chunming@hotmail.com
% URL: http://www.imagecomputing.org/~cmli/

phi=phi_0;
[vx, vy]=gradient(g); %gradiente de la función de bordes
for k=1:iter
    phi=NeumannBoundCond(phi);
    [phi_x,phi_y]=gradient(phi);
    s=sqrt(phi_x.^2 + phi_y.^2);
    smallNumber=1e-10;
    Nx=phi_x./(s+smallNumber); % añade un pequeño número positivo para
eliminar la posibilidad de división entre cero
    Ny=phi_y./(s+smallNumber);
    curvature=div(Nx,Ny);
    if strcmp(potentialFunction,'single-well')
```



```

        distRegTerm = 4*del2(phi)-curvature; % computa el término de
        distancia de regularización con potencial simple p1
    elseif strcmp(potentialFunction,'double-well');
        distRegTerm=distReg_p2(phi); % computa el término de distancia
        de regularización con potencial doble p2
    else
        disp('Error: Wrong choice of potential function. Please input the
        string "single-well" or "double-well" in the drlse_edge function.');
```

end

```

    diracPhi=Dirac(phi,epsilon); %Función delta de Dirac
    areaTerm=diracPhi.*g; % término de área ponderada

    edgeTerm=diracPhi.*(vx.*Nx+vy.*Ny) + diracPhi.*g.*curvature; %Término
    de longitud ponderada

    phi=phi + timestep*(mu*distRegTerm + lambda*edgeTerm +
    alfa*areaTerm); %Cálculo del funcional
end

function f = distReg_p2(phi)
% computa el término de distancia de regularización con potencial doble
    [phi_x,phi_y]=gradient(phi);
    s=sqrt(phi_x.^2 + phi_y.^2);
    a=(s>=0) & (s<=1);
    b=(s>1);
    ps=a.*sin(2*pi*s)/(2*pi)+b.*(s-1); % computa la derivada de primer orden
    del potencial doble
    dps=((ps~=0).*ps+(ps==0))./((s~=0).*s+(s==0));
    d_p(s)=p'(s)/s.
    f = div(dps.*phi_x - phi_x, dps.*phi_y - phi_y) + 4*del2(phi);

function f = div(nx,ny) %Función gradiente
    [nxx,junk]=gradient(nx);
    [junk,nyy]=gradient(ny);
```



```
f=nxx+nyy;
```

```
function f = Dirac(x, sigma) %Función delta de Dirac
```

```
f=(1/2/sigma)*(1+cos(pi*x/sigma));
```

```
b = (x<=sigma) & (x>=-sigma);
```

```
f = f.*b;
```

```
function g = NeumannBoundCond(f)
```

```
% Función que cumple las condiciones de contorno de Neumann
```

```
[nrow,ncol] = size(f);
```

```
g = f;
```

```
g([1 nrow],[1 ncol]) = g([3 nrow-2],[3 ncol-2]);
```

```
g([1 nrow],2:end-1) = g([3 nrow-2],2:end-1);
```

```
g(2:end-1,[1 ncol]) = g(2:end-1,[3 ncol-2]);
```

## Script ejecutable

```
% Author: Chunming Li, all rights reserved
```

```
% E-mail: lchunming@gmail.com
```

```
% li_chunming@hotmail.com
```

```
% URL: http://www.imagecomputing.org/~cmli/
```

```
%Se ha realizado una adaptación del código de Chunming Li aumentando las  
funciones de este con variaciones del valor del parámetro alpha,  
representando la evolución del snake en cada caso. Además, se tratan las  
imágenes dadas con ROI obteniendo únicamente su contorno y el contorno  
del snake para luego comprobar la fiabilidad del método representando un  
contorno sobre otro y dibujando la curva ROC por cada iteración de alpha.
```

```
clear all;
```

```
close all;
```

```
%Lectura de la imagen con ROI
```

```
imagen=imread('m5.png');
```

```
[m,n,p]=size(imagen);
```

```
imf=imagen(:,:,1:3); %Definición de los tres canales de la imagen
```



```
imshow(imf);
img=zeros(m,n); %Creación de una matriz de ceros
for i=1:m
    for j=1:n
        %Se asigna R,G,B a cada canal
        R=imf(:, :, 1);
        G=imf(:, :, 2);
        B=imf(:, :, 3);
        if(B(i,j)~=1 && B(i,j)~=255) %Condición de binarización y
            segmentación de contorno
                img(i,j)=1;
            else
                img(i,j)=0;
            end
        end
    end
end
imshow(img);
Img=dicomread('m5'); %lectura de la imagen
Img=double(Img(:, :, 1));
Img=imresize(Img, [256 256])
%% Configuración de parámetros
timestep=1; % time step
mu=0.2/timestep; % coeficiente del término de la distancia regularización,
R(phi)
iter_inner=50; %iteraciones internas
iter_outer=20; %iteraciones externas
lambda=5; % coeficiente de peso del término longitud L(phi)
i=1;
for alpha=-10:10 %Variación de los valores de alpha, coeficiente de peso del
término de área A(phi)
    alfa=alpha+1; % incremento de alpha en cada iteración
    epsilon=1.5; % parámetro que especifica el ancho de integral de línea de
la función delta de Dirac
```



```

sigma=1.8; % parámetro de desviación estándar en el filtro Gaussiano
H=fspecial('gaussian',20,sigma); % Filtro Gaussiano
Img_smooth=conv2(Img,H,'same'); % Convolución de la imagen filtrada
suavizada

[fx,fy]=gradient(Img_smooth); %Formulación del gradiente
f=fx.^2+fy.^2;
g=1./(1+f); % Función indicador de bordes
% Inicialización de LSF como una función binaria escalón
c0=2;
initialLSF = c0*ones(size(Img));
% Generar la región inicial R0 como rectángulos
initialLSF(124:158,94:122)=-c0;
initialLSF(145:157,76:95)=-c0;
phi=initialLSF;
figure(1);
mesh(-phi); % Para poder verlo mejor, se representa LSF gráficamente
hold on; contour(phi, [0,0], 'r','LineWidth',2);
title('Initial level set function');
view([-80 35]);
figure(2); %Representación en la imagen de los rectángulos definidos
imagesc(Img,[0, 255]); axis off; axis equal; colormap(gray); hold on;
contour(phi, [0,0], 'r');
title('Initial zero level contour');
pause(0.5);
potential=2;
if potential ==1
    potentialFunction = 'single-well'; % usa el potencial simple  $p_1(s)=0.5*(s-1)^2$ 
elseif potential == 2
    potentialFunction = 'double-well'; % usa el potencial doble
else
    potentialFunction = 'double-well'; % por defecto se usa el potencial
doble

```



```
end
% Inicio de la evolución del Level Set
for n=1:iter_outer
    phi = drlse_edge(phi, g, lambda, mu, alfa, epsilon, timestep, iter_inner,
    potentialFunction);
    if mod(n,2)==0
        figure(2);
        imagesc(lmg,[0, 255]); axis off; axis equal; colormap(gray);
        hold on; contour(phi, [0,0], 'r');
    end
end
% Refinamiento del contorno de Level Set cero a partir de promover la
evolución de la curva con alfa=0
alfa=0;
iter_refine = 10;
phi = drlse_edge(phi, g, lambda, mu, alfa, epsilon, timestep, iter_inner,
potentialFunction);
finalLSF=phi;
figure(2);
    imagesc(lmg,[0, 255]); axis off; axis equal; colormap(gray);
    hold on; contour(phi, [0,0], 'r');
hold on; contour(phi, [0,0], 'r');
%Representación de la curva del snake adaptada en la imagen, con el
%número de iteraciones especificadas
str=['Final zero level contour, ', num2str(iter_outer*iter_inner+iter_refine), '
iterations'];
title(str);
figure;
    mesh(-finalLSF); % Para poder ver mejor la evolución del LSF, se dibuja
en una gráfica, indicando también el número de iteraciones que han sido
necesarias
hold on; contour(phi, [0,0], 'r','LineWidth',2);
view([-80 35]);
```





```
str=['Final level set function, ', num2str(iter_outer*iter_inner+iter_refine), '
iterations'];
title(str);
axis on;
[nrow, ncol]=size(Img);
axis([1 ncol 1 nrow -5 5]);
set(gca,'ZTick',[-3:1:3]);
set(gca,'FontSize',14)
bwims=(finalLSF<0); %Declaración de la imagen con el contorno LSF
evolucionado
im=logical(img);
im2=bwpropfilt(im,'Area',1); %Extrae el área más grande de la imagen
con características comunes
imb=bwmorph(im2,'fill'); %Rellenado de huecos
bwim2=imfill(imb,'holes');%Contorno deseado relleno de la imagen ROI

%CURVA ROC
imTP=bwim2 & bwims; %Cálculo lógico de los TP: intersección
imFN=bwim2>imTP; %Cálculo lógico de los FN: Resta lógica imagen ROI
y los TP
imshow(imFN);
imshow(imTP);
imFP=bwims>imTP; %Cálculo lógico de los FP: Resta lógica imagen snake
obtenida y los TP
imshow(imFP);
imTN= ~(imTP | imFN | imFP);%Cálculo lógico de los TN
imshow(imTN);
IM= 1*imTP + 2*imFN + 3*imFP;
%cyan TP, rojo FN, negro FP, blanco TN
map=[0 1 1;1 0 0;0 0 0];
cmap=colormap(map);
IMcolor = label2rgb(IM, cmap, 'w');
imshow(IMcolor);
```



```

title(['alpha=',num2str(alpha)]);%Representación de la superposición de
imágenes para poder visualizar los TP,TN,FP,FN
TP(i)=sum(imTP(:)); %suma de los píxeles correspondientes a los TP
TN(i)=sum(imTN(:));
FP(i)=sum(imFP(:));
FN(i)=sum(imFN(:));
sens(i)=TP(i)/(TP(i)+FN(i)); %Cálculo de la sensibilidad
espec(i)=TN(i)/(TN(i)+FP(i));%Cálculo de la especificidad
if((1-espec(i)) ==0 && sens(i) == 0) %Condición para que no haga más
iteraciones de las necesarias cuando llegue a 0
    break;
end
i=i+1;
end
sens=sens(:);
espec=espec(:);
espec2=1-espec;
plot(espec2,sens,'o-'); %Dibujo de la gráfica ROC, en el eje x 1- Especificidad,
en el eje y sensibilidad
title('Curva ROC');
ylabel('Sensibilidad');
xlabel('Especificidad');
axis square;
xlim([0 1]); ylim([0 1]); %Dibujar la gráfica del mismo tamaño
%Mostrar una tabla con los parámetros necesarios
TP=TP';
FP=FP';
TN=TN';
FN=FN';
h=alpha(end); %último elemento del array alpha
alpha=-10:h; %definir el tamaño del array alpha
alpha=alpha';
tabla=table(alpha,TP,TN,FP,FN,sens,espec2)
    
```



## Cálculo de AUC

```
sens=flipud(sens);%comando que le da la vuelta al array, ya que se quiere
calcular el área desde 0 hasta 1, y el vector está definido de 1 hasta 0
espec2=flipud(espec2);
n=length(espec2);
h = n-1;
AUC=0; %Inicialización
xm=zeros(n); %Creación de matrices de ceros de tamaño igual a la longitud
de los vectores
inc=zeros(n);
for i=2:n
    xm(i)= (espec2(i)+espec2(i-1))/2; %Cálculo de la media entre dos puntos
    if(i<=h)
        inc(i)=xm(i)-xm(i-1); %Cálculo del incremento del intervalo entre Xm
        AUC=AUC+(inc(i)*sens(i)); %Cálculo de AUC
    end
end
inc(n)=1-xm(n-1);%Definición de la última posición
inc(1)=xm(1); %Definición de la primera posición

AUC = AUC + (inc(1)*sens(1)); %Suma al resultado la primera posición
AUC = AUC + (inc(n)*sens(n)); %Suma al resultado la última posición
```

## Cálculo del valor óptimo de alpha

```
%Se leen las tablas de los excel creados
t4=xlsread('tabledata4.xlsx');
t5=xlsread('tabledata5.xlsx');
t6=xlsread('tabledata6.xlsx');
t7=xlsread('tabledata7.xlsx');
t8=xlsread('tabledata8.xlsx');
t10=xlsread('tabledata10.xlsx');
```



```
t11=xlsread('tabledata11.xlsx');
t12=xlsread('tabledata12.xlsx');
t13=xlsread('tabledata13.xlsx');
t14=xlsread('tabledata14.xlsx');
t15=xlsread('tabledata15.xlsx');
t16=xlsread('tabledata16.xlsx');
t17=xlsread('tabledata17.xlsx');
tm1=xlsread('tabledatam1.xlsx');
tm2=xlsread('tabledatam2.xlsx');
tm3=xlsread('tabledatam3.xlsx');
tm4=xlsread('tabledatam4.xlsx');
tm5=xlsread('tabledatam5.xlsx');

% Se obtiene la media de los valores de sensibilidad para cada iteración de
% %alpha
m=21;
n=7;
suma=zeros(m,n);
media=zeros(m,n);
for i=1:m
[m,n]= size(t4);
    if(m>=i)
        suma(i,6)=suma(i,6)+t4(i,6);
    end
[m,n]= size(t5);
    if(m>=i)
        suma(i,6)=suma(i,6)+t5(i,6);
    end
[m,n]= size(t6);
    if(m>=i)
        suma(i,6)=suma(i,6)+t6(i,6);
    end
[m,n]= size(t7);
```



```
if(m>=i)
    suma(i,6)=suma(i,6)+t7(i,6);
end
[m,n]= size(t8);
if(m>=i)
    suma(i,6)=suma(i,6)+t8(i,6);
end
[m,n]= size(t10);
if(m>=i)
    suma(i,6)=suma(i,6)+t10(i,6);
end
[m,n]= size(t11);
if(m>=i)
    suma(i,6)=suma(i,6)+t11(i,6);
end
[m,n]= size(t12);
if(m>=i)
    suma(i,6)=suma(i,6)+t12(i,6);
end
[m,n]= size(t13);
if(m>=i)
    suma(i,6)=suma(i,6)+t13(i,6);
end
[m,n]= size(t14);
if(m>=i)
    suma(i,6)=suma(i,6)+t14(i,6);
end
[m,n]= size(t15);
if(m>=i)
    suma(i,6)=suma(i,6)+t15(i,6);
end
[m,n]= size(t16);
if(m>=i)
```



```
        suma(i,6)=suma(i,6)+t16(i,6);
    end
    [m,n]= size(t17);
    if(m>=i)
        suma(i,6)=suma(i,6)+t17(i,6);
    end
    [m,n]= size(tm1);
    if(m>=i)
        suma(i,6)=suma(i,6)+tm1(i,6);
    end
    [m,n]= size(tm2);
    if(m>=i)
        suma(i,6)=suma(i,6)+tm2(i,6);
    end
    [m,n]= size(tm3);
    if(m>=i)
        suma(i,6)=suma(i,6)+tm3(i,6);
    end
    [m,n]= size(tm4);
    if(m>=i)
        suma(i,6)=suma(i,6)+tm4(i,6);
    end
    [m,n]= size(tm5);
    if(m>=i)
        suma(i,6)=suma(i,6)+tm5(i,6);
    end
    media(i,6)=suma(i,6)/18;
end

m=21;
n=7;
suma2=zeros(m,n);
media2=zeros(m,n);
```



```
for i=1:m
[m,n]= size(t4);
    if(m>=i)
        suma2(i,7)=suma2(i,7)+t4(i,7);
    end
[m,n]= size(t5);
    if(m>=i)
        suma2(i,7)=suma2(i,7)+t5(i,7);
    end
[m,n]= size(t6);
    if(m>=i)
        suma2(i,7)=suma2(i,7)+t6(i,7);
    end
[m,n]= size(t7);
    if(m>=i)
        suma2(i,7)=suma2(i,7)+t7(i,7);
    end
[m,n]= size(t8);
    if(m>=i)
        suma2(i,7)=suma2(i,7)+t8(i,7);
    end
[m,n]= size(t10);
    if(m>=i)
        suma2(i,7)=suma2(i,7)+t10(i,7);
    end
[m,n]= size(t11);
    if(m>=i)
        suma2(i,7)=suma2(i,7)+t11(i,7);
    end
[m,n]= size(t12);
    if(m>=i)
        suma2(i,7)=suma2(i,7)+t12(i,7);
    end
end
```



```
[m,n]= size(t13);  
if(m>=i)  
    suma2(i,7)=suma2(i,7)+t13(i,7);  
end  
[m,n]= size(t14);  
if(m>=i)  
    suma2(i,7)=suma2(i,7)+t14(i,7);  
end  
[m,n]= size(t15);  
if(m>=i)  
    suma2(i,7)=suma2(i,7)+t15(i,7);  
end  
[m,n]= size(t16);  
if(m>=i)  
    suma2(i,7)=suma2(i,7)+t16(i,7);  
end  
[m,n]= size(t17);  
if(m>=i)  
    suma2(i,7)=suma2(i,7)+t17(i,7);  
end  
[m,n]= size(tm1);  
if(m>=i)  
    suma2(i,7)=suma2(i,7)+tm1(i,7);  
end  
[m,n]= size(tm2);  
if(m>=i)  
    suma2(i,7)=suma2(i,7)+tm2(i,7);  
end  
[m,n]= size(tm3);  
if(m>=i)  
    suma2(i,7)=suma2(i,7)+tm3(i,7);  
end  
[m,n]= size(tm4);
```





```
if(m>=i)
    suma2(i,7)=suma2(i,7)+tm4(i,7);
end
[m,n]= size(tm5);
if(m>=i)
    suma2(i,7)=suma2(i,7)+tm5(i,7);
end
media2(i,7)=suma2(i,7)/18;
end

media=media(:,6);
media2=media2(:,7);
plot(media2,media,'o-'); %Representación de la curva ROC
title('Media aritmética para cada alpha');
ylabel('Sensibilidad');
xlabel('Especificidad');
axis square;
xlim([0 1]); %Poner la grafica del mismo tamaño
ylim([0 1]);
alpha=-10:10;
alpha=alpha';
sens=media;
espec2=media2;
tabla=table(alpha,sens,espec2); %Obtención de la tabla
```



## ANEXO B. IMÁGENES DE REFERENCIA CON ROI

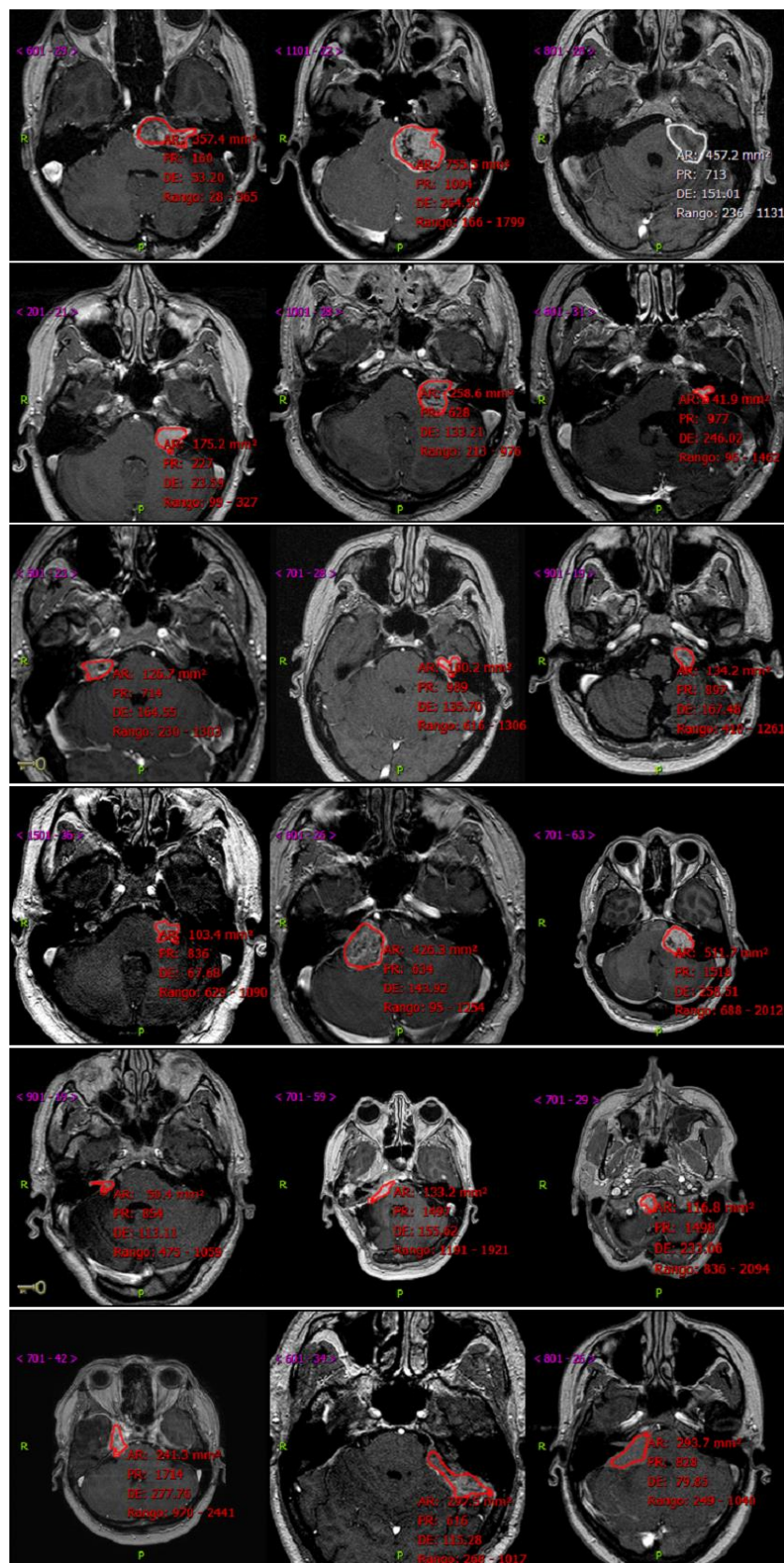


Figura 41. Imágenes de referencia con ROI

## ANEXO C. REPRESENTACIÓN DE LA CURVA ROC

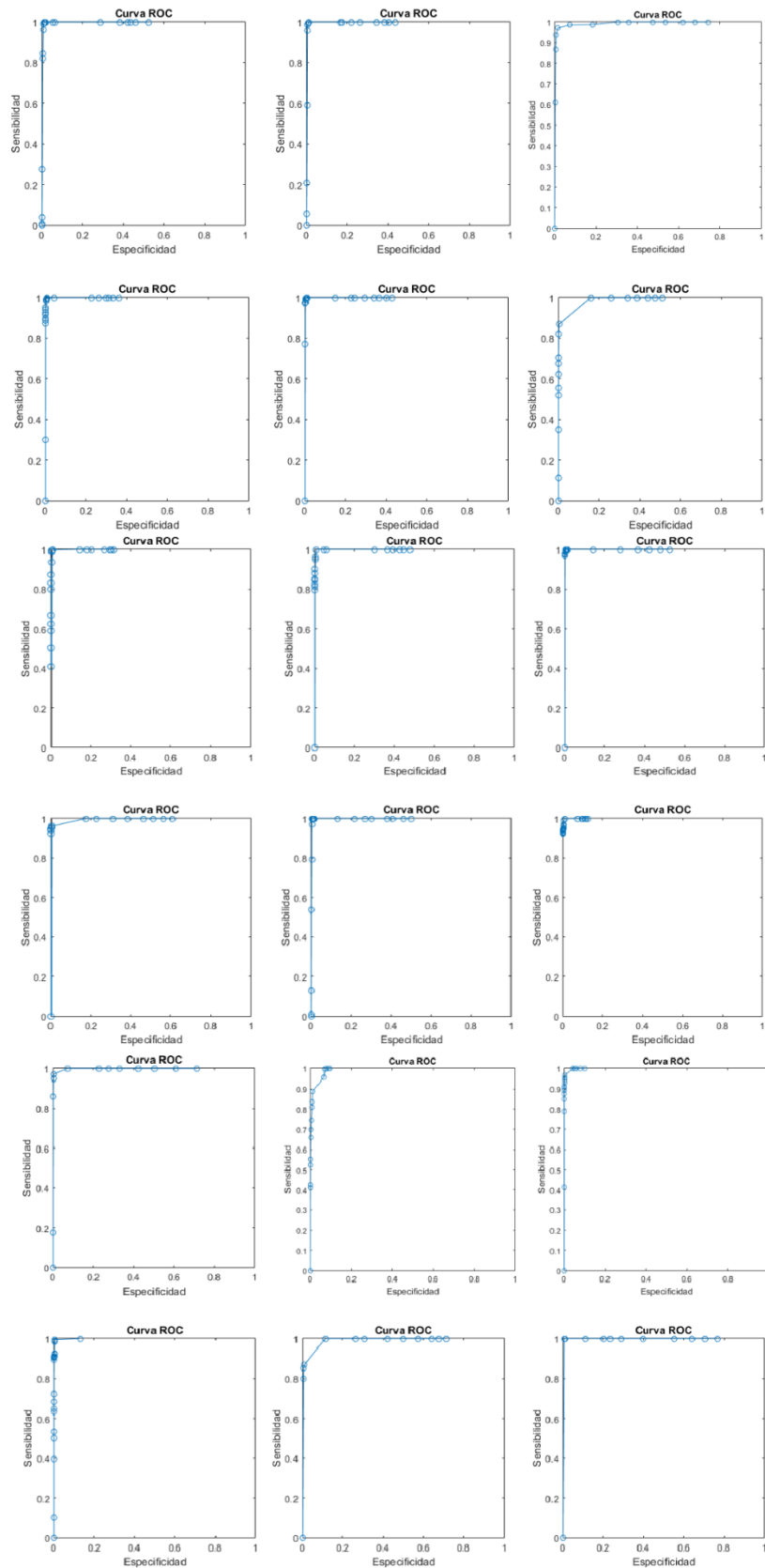


Figura 42. Curvas ROC obtenidas de cada imagen



## ANEXO D. VALORES DE AUC

*Tabla 4. Valores de AUC obtenidos para cada imagen*

AUC= 0.9988	AUC = 0.9989	AUC = 0.9975
AUC = 0.9998	AUC = 0.9998	AUC= 0.9997
AUC = 0.9994	AUC = 0.9997	AUC = 1.0000
AUC = 0.9998	AUC =0.9993	AUC = 0.9998
AUC = 0.9997	AUC = 0.9960	AUC = 0.9997
AUC = 0.9995	AUC = 0.9990	AUC= 0.9998



## ANEXO E. TABLAS OBTENIDAS CON LOS PARÁMETROS NECESARIOS PARA LA CURVA ROC

alpha	TP	TN	FP	FN	sens	espec2
-10	1014	30766	33756	0	1	0,52317039
-9	1014	34731	29791	0	1	0,46171848
-8	1014	36384	28138	0	1	0,43609931
-7	1014	37397	27125	0	1	0,42039924
-6	1014	39850	24672	0	1	0,3823812
-5	1014	46091	18431	0	1	0,28565451
-4	1014	60516	4006	0	1	0,06208735
-3	1014	61109	3413	0	1	0,05289669
-2	1014	63456	1066	0	1	0,0165215
-1	1014	63642	880	0	1	0,01363876
0	1014	63752	770	0	1	0,01193391
1	1004	64094	428	10	0,99013807	0,0066334
2	975	64166	356	39	0,96153846	0,0055175
3	857	64234	288	157	0,84516765	0,00446359
4	831	64312	210	183	0,81952663	0,0032547
5	277	64433	89	737	0,27317554	0,00137937
6	38	64506	16	976	0,03747535	0,00024798
7	4	64510	12	1010	0,00394477	0,00018598
8	4	64510	12	1010	0,00394477	0,00018598
9	4	64510	12	1010	0,00394477	0,00018598
10	0	64522	0	1014	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	1625	36061	27850	0	1	0,43576223
-9	1625	38012	25899	0	1	0,40523541
-8	1625	39464	24447	0	1	0,38251631
-7	1625	42020	21891	0	1	0,3425232
-6	1625	47060	16851	0	1	0,26366353
-5	1625	49824	14087	0	1	0,22041589
-4	1625	52732	11179	0	1	0,17491512



-3	1625	53181	10730	0	1	0,16788972
-2	1625	63210	701	0	1	0,01096838
-1	1619	63429	482	6	0,99630769	0,00754174
0	1617	63457	454	8	0,99507692	0,00710363
1	1604	63612	299	21	0,98707692	0,00467838
2	1559	63646	265	66	0,95938462	0,00414639
3	959	63743	168	666	0,59015385	0,00262866
4	338	63835	76	1287	0,208	0,00118915
5	91	63911	0	1534	0,056	0
6	0	63911	0	1625	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	1013	16648	47875	0	1	0,74198348
-9	1013	20611	43912	0	1	0,68056352
-8	1013	24386	40137	0	1	0,62205725
-7	1013	29936	34587	0	1	0,53604141
-6	1013	33778	30745	0	1	0,47649675
-5	1013	41473	23050	0	1	0,35723695
-4	1013	44831	19692	0	1	0,3051935
-3	1001	52669	11854	12	0,988154	0,18371743
-2	999	59882	4641	14	0,98617966	0,07192784
-1	984	63611	912	29	0,97137216	0,01413449
0	950	64060	463	63	0,93780849	0,00717574
1	878	64234	289	135	0,86673248	0,00447902
2	620	64357	166	393	0,61204344	0,00257273
3	0	64523	0	1013	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	518	41458	23560	0	1	0,36236119
-9	518	43409	21609	0	1	0,33235412
-8	518	44627	20391	0	1	0,31362084
-7	518	45619	19399	0	1	0,29836353
-6	518	48058	16960	0	1	0,26085084
-5	518	50246	14772	0	1	0,22719862
-4	518	62221	2797	0	1	0,04301886



-3	517	64623	395	1	0,9980695	0,00607524
-2	516	64666	352	2	0,996139	0,00541389
-1	514	64774	244	4	0,99227799	0,00375281
0	513	64851	167	5	0,99034749	0,00256852
1	509	64864	154	9	0,98262548	0,00236857
2	493	64957	61	25	0,95173745	0,0009382
3	487	64962	56	31	0,94015444	0,0008613
4	480	64964	54	38	0,92664093	0,00083054
5	474	64968	50	44	0,91505792	0,00076902
6	466	64973	45	52	0,8996139	0,00069212
7	460	64977	41	58	0,88803089	0,00063059
8	452	64981	37	66	0,87258687	0,00056907
9	156	65011	7	362	0,3011583	0,00010766
10	0	65018	0	518	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	829	36998	27709	0	1	0,4282226
-9	829	38795	25912	0	1	0,40045126
-8	829	41182	23525	0	1	0,3635619
-7	829	42762	21945	0	1	0,33914414
-6	829	45770	18937	0	1	0,29265767
-5	829	48880	15827	0	1	0,24459487
-4	829	49961	14746	0	1	0,22788879
-3	829	55173	9534	0	1	0,14734109
-2	829	64045	662	0	1	0,01023073
-1	829	64215	492	0	1	0,00760351
0	826	64314	393	3	0,99638118	0,00607353
1	825	64377	330	4	0,99517491	0,00509991
2	820	64400	307	9	0,98914355	0,00474446
3	811	64584	123	18	0,97828709	0,00190088
4	808	64595	112	21	0,97466828	0,00173088
5	640	64644	63	189	0,77201448	0,00097362
6	0	64707	0	829	0	0





alpha	TP	TN	FP	FN	sens	espec2
-10	169	31986	33381	0	1	0,51067052
-9	169	34232	31135	0	1	0,47631068
-8	169	36729	28638	0	1	0,43811097
-7	169	40026	25341	0	1	0,38767268
-6	169	43149	22218	0	1	0,33989628
-5	169	48468	16899	0	1	0,25852494
-4	169	55041	10326	0	1	0,15796962
-3	147	65191	176	22	0,86982249	0,00269249
-2	139	65297	70	30	0,82248521	0,00107088
-1	119	65337	30	50	0,70414201	0,00045895
0	114	65340	27	55	0,67455621	0,00041305
1	105	65343	24	64	0,62130178	0,00036716
2	105	65348	19	64	0,62130178	0,00029067
3	94	65353	14	75	0,55621302	0,00021418
4	88	65354	13	81	0,52071006	0,00019888
5	59	65360	7	110	0,34911243	0,00010709
6	19	65366	1	150	0,11242604	1,5298E-05
7	0	65367	0	169	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	423	44606	20507	0	1	0,31494479
-9	423	45180	19933	0	1	0,30612934
-8	423	45748	19365	0	1	0,29740605
-7	423	47678	17435	0	1	0,26776527
-6	423	51949	13164	0	1	0,20217161
-5	423	53547	11566	0	1	0,17762966
-4	423	55646	9467	0	1	0,14539339
-3	422	64571	542	1	0,99763593	0,00832399
-2	422	64680	433	1	0,99763593	0,00664998
-1	420	64786	327	3	0,9929078	0,00502204
0	420	64796	317	3	0,9929078	0,00486846
1	418	64827	286	5	0,98817967	0,00439236
2	397	64983	130	26	0,93853428	0,00199653
3	370	65011	102	53	0,87470449	0,00156651



4	352	65023	90	71	0,8321513	0,00138221
5	338	65027	86	85	0,79905437	0,00132078
6	282	65044	69	141	0,66666667	0,0010597
7	265	65054	59	158	0,62647754	0,00090612
8	249	65064	49	174	0,58865248	0,00075254
9	212	65081	32	211	0,50118203	0,00049145
10	173	65090	23	250	0,40898345	0,00035323

alpha	TP	TN	FP	FN	sens	espec2
-10	253	34126	31157	0	1	0,47726054
-9	253	36114	29169	0	1	0,44680851
-8	253	37570	27713	0	1	0,42450561
-7	253	39676	25607	0	1	0,39224607
-6	253	41461	23822	0	1	0,36490357
-5	253	45631	19652	0	1	0,30102783
-4	253	61344	3939	0	1	0,0603373
-3	253	62163	3120	0	1	0,04779192
-2	253	64782	501	0	1	0,00767428
-1	244	65013	270	9	0,96442688	0,00413584
0	241	65081	202	12	0,95256917	0,00309422
1	228	65213	70	25	0,90118577	0,00107225
2	223	65217	66	30	0,88142292	0,00101098
3	216	65229	54	37	0,85375494	0,00082717
4	214	65235	48	39	0,8458498	0,00073526
5	209	65241	42	44	0,82608696	0,00064335
6	206	65245	38	47	0,81422925	0,00058208
7	201	65248	35	52	0,7944664	0,00053613
8	0	65283	0	253	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	294	30690	34552	0	1	0,5295975
-9	294	34020	31222	0	1	0,47855676
-8	294	37819	27423	0	1	0,4203274
-7	294	41285	23957	0	1	0,36720211
-6	294	47184	18058	0	1	0,27678489



-5	294	56019	9223	0	1	0,14136599
-4	294	64204	1038	0	1	0,01591
-3	294	64571	671	0	1	0,01028479
-2	294	64717	525	0	1	0,00804696
-1	294	64892	350	0	1	0,00536464
0	294	64990	252	0	1	0,00386254
1	291	65051	191	3	0,98979592	0,00292756
2	290	65119	123	4	0,98639456	0,00188529
3	288	65187	55	6	0,97959184	0,00084302
4	287	65190	52	7	0,97619048	0,00079703
5	285	65193	49	9	0,96938776	0,00075105
6	0	65242	0	294	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	1322	32184	32030	0	1	0,49880088
-9	1322	34501	29713	0	1	0,46271841
-8	1322	38016	26198	0	1	0,40797957
-7	1322	39745	24469	0	1	0,38105398
-6	1322	44855	19359	0	1	0,30147631
-5	1322	46900	17314	0	1	0,26962968
-4	1322	50414	13800	0	1	0,21490641
-3	1322	55745	8469	0	1	0,13188713
-2	1322	63276	938	0	1	0,01460741
-1	1322	63541	673	0	1	0,01048058
0	1322	63689	525	0	1	0,00817579
1	1320	63960	254	2	0,99848714	0,00395552
2	1281	64022	192	41	0,96898638	0,00299
3	1050	64055	159	272	0,79425113	0,0024761
4	710	64143	71	612	0,53706505	0,00110568
5	169	64191	23	1153	0,12783661	0,00035818
6	11	64212	2	1311	0,00832073	3,1146E-05
7	0	64214	0	1322	0	0



alpha	TP	TN	FP	FN	sens	espec2
-10	368	25463	39705	0	1	0,60927142
-9	368	28330	36838	0	1	0,56527744
-8	368	31918	33250	0	1	0,51021974
-7	368	34903	30265	0	1	0,46441505
-6	368	40121	25047	0	1	0,38434508
-5	368	44905	20263	0	1	0,31093481
-4	368	50276	14892	0	1	0,22851706
-3	368	53809	11359	0	1	0,17430334
-2	354	64946	222	14	0,96195652	0,00340658
-1	354	64978	190	14	0,96195652	0,00291554
0	352	65047	121	16	0,95652174	0,00185674
1	349	65055	113	19	0,94836957	0,00173398
2	345	65066	102	23	0,9375	0,00156519
3	338	65107	61	30	0,91847826	0,00093604
4	0	65168	0	368	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	622	56929	7985	0	1	0,1230089
-9	622	57506	7408	0	1	0,11412022
-8	622	57742	7172	0	1	0,11048464
-7	622	57891	7023	0	1	0,1081893
-6	622	58524	6390	0	1	0,09843793
-5	622	58709	6205	0	1	0,09558801
-4	622	60094	4820	0	1	0,07425209
-3	622	64162	752	0	1	0,01158456
-2	617	64581	333	5	0,99196141	0,00512986
-1	603	64659	255	19	0,96945338	0,00392827
0	599	64763	151	23	0,96302251	0,00232615
1	593	64772	142	29	0,95337621	0,00218751
2	591	64785	129	31	0,95016077	0,00198724
3	590	64792	122	32	0,94855305	0,00187941
4	587	64799	115	35	0,9437299	0,00177157
5	585	64803	111	37	0,94051447	0,00170995
6	584	64807	107	38	0,93890675	0,00164833



7	581	64807	107	41	0,9340836	0,00164833
8	576	64807	107	46	0,92604502	0,00164833
9	574	64812	102	48	0,92282958	0,00157131
10	573	64817	97	49	0,92122186	0,00149428

alpha	TP	TN	FP	FN	sens	espec2
-10	137	18866	46533	0	1	0,71152464
-9	137	25528	39871	0	1	0,60965764
-8	137	32528	32871	0	1	0,50262236
-7	137	37765	27634	0	1	0,42254469
-6	137	43747	21652	0	1	0,3310754
-5	137	47231	18168	0	1	0,27780241
-4	137	50496	14903	0	1	0,2278781
-3	137	60702	4697	0	1	0,07182067
-2	133	65173	226	4	0,97080292	0,00345571
-1	130	65259	140	7	0,94890511	0,00214071
0	118	65327	72	19	0,86131387	0,00110093
1	24	65382	17	113	0,17518248	0,00025994
2	0	65399	0	137	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	208	58849	6479	0	1	0,09917646
-9	208	60140	5188	0	1	0,07941465
-8	208	61614	3714	0	1	0,05685158
-7	208	61962	3366	0	1	0,05152461
-6	208	62393	2935	0	1	0,04492714
-5	201	65059	269	7	0,96634615	0,00411768
-4	198	65083	245	10	0,95192308	0,00375031
-3	198	65157	171	10	0,95192308	0,00261756
-2	196	65215	113	12	0,94230769	0,00172973
-1	194	65228	100	14	0,93269231	0,00153074
0	190	65275	53	18	0,91346154	0,00081129
1	188	65285	43	20	0,90384615	0,00065822
2	186	65289	39	22	0,89423077	0,00059699
3	182	65291	37	26	0,875	0,00056637



4	177	65293	35	31	0,85096154	0,00053576
5	164	65297	31	44	0,78846154	0,00047453
6	86	65304	24	122	0,41346154	0,00036738
7	0	65328	0	208	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	205	59162	6169	0	1	0,09442684
-9	205	59711	5620	0	1	0,08602348
-8	205	60288	5043	0	1	0,07719153
-7	204	60742	4589	1	0,99512195	0,0702423
-6	196	61126	4205	9	0,95609756	0,06436454
-5	182	64570	761	23	0,88780488	0,01164838
-4	171	64683	648	34	0,83414634	0,00991872
-3	166	64734	597	39	0,8097561	0,00913808
-2	152	64981	350	53	0,74146341	0,00535733
-1	143	65117	214	62	0,69756098	0,00327563
0	135	65184	147	70	0,65853659	0,00225008
1	113	65239	92	92	0,55121951	0,00140821
2	107	65254	77	98	0,52195122	0,00117861
3	87	65277	54	118	0,42439024	0,00082656
4	84	65289	42	121	0,4097561	0,00064288
5	0	65330	1	205	0	1,5307E-05
6	0	65331	0	205	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	310	56760	8466	0	1	0,12979487
-9	308	64924	302	2	0,99354839	0,00463006
-8	306	64942	284	4	0,98709677	0,00435409
-7	305	64962	264	5	0,98387097	0,00404747
-6	305	64986	240	5	0,98387097	0,00367951
-5	288	65070	156	22	0,92903226	0,00239168
-4	286	65081	145	24	0,92258065	0,00222304
-3	283	65105	121	27	0,91290323	0,00185509
-2	283	65117	109	27	0,91290323	0,00167111
-1	282	65126	100	28	0,90967742	0,00153313



0	281	65135	91	29	0,90645161	0,00139515
1	278	65143	83	32	0,89677419	0,0012725
2	224	65175	51	86	0,72258065	0,0007819
3	212	65184	42	98	0,68387097	0,00064392
4	201	65198	28	109	0,6483871	0,00042928
5	196	65201	25	114	0,63225806	0,00038328
6	165	65207	19	145	0,53225806	0,00029129
7	156	65213	13	154	0,50322581	0,00019931
8	123	65216	10	187	0,39677419	0,00015331
9	31	65224	2	279	0,1	3,0663E-05
10	0	65226	0	310	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	1008	18520	46008	0	1	0,71299281
-9	1008	20911	43617	0	1	0,67593913
-8	1008	23075	41453	0	1	0,6424033
-7	1008	27354	37174	0	1	0,576091
-6	1008	32292	32236	0	1	0,49956608
-5	1008	37370	27158	0	1	0,42087156
-4	1008	44632	19896	0	1	0,30833127
-3	1008	47686	16842	0	1	0,26100298
-2	1008	57240	7288	0	1	0,11294322
-1	876	64027	501	132	0,86904762	0,00776407
0	858	64219	309	150	0,85119048	0,00478862
1	807	64298	230	201	0,80059524	0,00356434
2	0	64528	0	1008	0	0

alpha	TP	TN	FP	FN	sens	espec2
-10	654	14947	49935	0	1	0,76962794
-9	654	19188	45694	0	1	0,70426312
-8	654	23148	41734	0	1	0,64322925
-7	654	28959	35923	0	1	0,55366666
-6	654	39085	25797	0	1	0,39759872
-5	654	46013	18869	0	1	0,29082026
-4	654	49794	15088	0	1	0,23254524



-3	654	51684	13198	0	1	0,20341543
-2	654	57852	7030	0	1	0,10835054
-1	654	64262	620	0	1	0,00955581
0	654	64526	356	0	1	0,00548688
1	0	64882	0	654	0	0